

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”**

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

“На правах рукопису”

УДК 004.9

“До захисту допущено”

Науковий керівник кафедри

_____ І. А. Дичка

“ ____ ” _____ 2019 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 121 Інженерія програмного забезпечення

**на тему: “Спосіб та програмне забезпечення ауралізації передачі
повітряного шуму”**

Виконав:

студент II курсу, групи КП-81мп
Глінський Владислав Васильович

Керівник:

Доцент кафедри ПЗКС, к.т.н., доцент,
Сулема Є. С.

Консультант з нормоконтролю:

Доцент кафедри ПЗКС, к.т.н., доцент,
Онай М. В.

Рецензент:

В.о. завідувача кафедри ММСА ІПСА, к.т.н., доцент,
Тимощук О. Л.

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних
посилань.

Студент _____

Київ – 2019 року

Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет прикладної математики
Кафедра програмного забезпечення комп’ютерних систем

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою

Спеціальність (спеціалізація) – 121 “Інженерія програмного забезпечення”
 (“Програмне забезпечення систем”)

ЗАТВЕРДЖУЮ

Науковий керівник кафедри

_____ І. А. Дичка

“ _____ ” _____ 2018 р.

ЗАВДАННЯ
на магістерську дисертацію студенту

Глінському Владиславу Васильовичу

1. Тема дисертації: “Спосіб та програмне забезпечення ауралізації передачі повітряного шуму”, науковий керівник дисертації: Сулема Євгенія Станіславівна, к.т.н., доцент, затверджені наказом по університету від “13” листопада 2019 р. № 3895-с.
2. Термін подання студентом завершеної дисертації “11” грудня 2019 р.
3. Об’єкт дослідження: процес програмного перетворення аудіосигналу.
4. Предмет дослідження: способи та алгоритми програмного моделювання акустичного простору.
5. Перелік задач, які потрібно вирішити:
 - провести аналіз існуючих методів моделювання акустичного простору;
 - розробити спосіб програмного відтворення поширення повітряного шуму в будівлі;
 - розробити програмний продукт, який реалізує розроблений спосіб.
6. Орієнтовний перелік ілюстративного матеріалу:
 - основні процеси ауралізації;
 - архітектура програмного проекту;
 - алгоритм пошуку всіх шляхів між двома вершинами неорієнтованого мультиграфа;
 - діаграма варіантів використання;
 - презентація.

7. Орієнтовний перелік публікацій:

- Тези доповіді “Спрощений спосіб ауралізації передачі повітряного шуму” – XII наукова конференція магістрантів та аспірантів «Прикладна математика та комп’ютинг» (ПМК-2019).

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Онай М. В., к.т.н., доцент		

9. Дата видачі завдання “15” жовтня 2018 р.

Календарний план

№ з/п	Назва етапу виконання магістерської дисертації	Термін виконання етапу дисертації	Примітка
1.	Грунтовне ознайомлення з предметною галуззю	14.11.2018	
2.	Визначення структури магістерської дисертації; вивчення літератури, пошук додаткової літератури, патентний пошук	28.12.2018	
3.	Робота над першим розділом магістерської дисертації; проведення наукового дослідження	15.03.2019	
4.	Проведення наукового дослідження; формулювання способу; робота над другим розділом магістерської дисертації	30.05.2019	
5.	Робота над створенням програмної реалізації розробленого способу та третім розділом магістерської дисертації	20.10.2019	
6.	Робота над четвертим та п’ятим розділом магістерської дисертації.	20.02.2019	
7.	Завершення роботи над основною частиною магістерської дисертації; підготовка ілюстративного матеріалу; підготовка матеріалів доповіді на конференції ПМК-2019	10.11.2019	
8.	Оформлення текстової та графічної частини магістерської дисертації	02.12.2019	

Студент

В. В. Глінський

Науковий керівник дисертації

Є. С. Сулема

РЕФЕРАТ

Актуальність. Офісні та житлові приміщення часто розміщені поряд з джерелами шуму різних типів і потребують звукоізоляції, яка здатна знизити максимальний рівень шуму в них до прийняттого. Неефективна звукоізоляція стає причиною негативного впливу шуму на здоров'я та продуктивність людей, порушення приватності та витоку інформації, а також знижує вартість приміщень. Як наслідок, є необхідність оцінювати ефективність шумоізоляції на етапі проектування або реконструкції приміщень.

В архітектурному проектуванні опис акустичних властивостей спроектованого оточення – приміщення або будівлі – подається певними числовими величинами та візуально, наприклад, за допомогою теплокарт. Створення можливості на слух оцінити вплив конструктивних елементів оточення на звук від наявних джерел шуму надасть наочності процесу проектування акустики приміщень і дозволить продемонструвати результати проектування користувачам-неспеціалістам. Для реалізації такої можливості пропонується розробити спосіб ауралізації поширення повітряного шуму та його програмну реалізацію. Ауралізацією називають сукупність способів відтворення звукових полів на основі числових даних, отриманих шляхом моделювання, вимірювання чи синтезу. Їх часто застосовують для створення інтерактивних оточень у засобах віртуальної реальності, наприклад, навчальних та тренувальних програмах.

Об'єктом дослідження є процес програмного перетворення аудіосигналу.

Предметом дослідження є способи та алгоритми програмного моделювання акустичного простору.

Метою дослідження є розроблення способу відтворення характеру зміни звуку в процесі його поширення в межах множини кімнат від джерел до слухача, а також створення програмної реалізації способу.

Методи дослідження. В роботі використовуються методи програмної інженерії, математичного моделювання та емпіричні методи.

Наукова новизна роботи полягає у розробленні способу ауралізації поширення повітряного шуму в будівлі, перевагою якого є простота реалізації та висока ефективність.

Розроблений спосіб може використовуватися для перетворення аудіосигналу в реальному часі з мінімально можливою втратою реалістичності звучання. Для зменшення кількості обчислень враховуються лише прямі шляхи поширення шуму без виникнення дифракції та реверберації, а перегородки-випромінювачі звуку вважаються вторинними точковими джерелами.

Практична цінність отриманих в роботі результатів полягає в тому, що програмна реалізація запропонованого способу є засобом, який дозволяє на слух оцінювати вплив конструктивних елементів будівлі на звукоізоляцію його приміщень.

Апробація роботи. Основні положення і результати роботи доповідалися та обговорювалися на XII науковій конференції магістрантів та аспірантів “Прикладна математика та комп’ютинг” ПМК-2019.

Структура та обсяг роботи. Магістерська дисертація складається з вступу, п’яти розділів, висновків та додатків.

У вступі надано загальну характеристику роботи, виконано оцінку сучасного стану проблеми, обґрунтовано актуальність напрямку дослідження.

У першому розділі розглянуто загальні підходи та технічні засоби, що використовуються в ауралізації, наводяться приклади методів, які типово застосовуються для симуляції акустики приміщень.

У другому розділі сформульовано науково-інноваційну задачу, проведено аналіз математичних моделей поширення звуку в будівлях та розглянуто підходи до створення цифрових фільтрів для оброблення

аудіосигналу, запропоновано спосіб ауралізації поширення повітряного шуму в межах множини кімнат будівлі від джерел до слухача.

У третьому розділі проведено виявлення та аналіз вимог до програмної реалізації розробленого способу ауралізації – окремої прикладної програми, побудованої з використанням програмного рушія Unity. Наводяться архітектура та ключові алгоритми програмного продукту.

У четвертому розділі описана методика тестування програмного продукту для забезпечення відповідності сформульованим вимогам, створено набір тестових даних та моделей оточення, виконано аналіз отриманих результатів та встановлено напрямки вдосконалення.

У п'ятому розділі з метою оцінки комерційної життєздатності програмної реалізації розробленого способу було розглянуто можливість створення стартап-проєкту на її основі. Побудовано бізнес-модель та наведено показники потенційної прибутковості.

У висновках проаналізовано отримані результати роботи.

У додатках наведено фрагменти вихідного коду програмної реалізації способу, копії графічних матеріалів та слайдів презентації.

Робота виконана на 84 аркушах, містить 2 додатки та посилання на список використаних літературних джерел з 34 найменувань. У роботі наведено 23 рисунки та 7 таблиць.

Ключові слова: ауралізація, моделювання звукових полів, акустика, поширення звуку, шумоізоляція, звукоізоляція, Unity.

ABSTRACT

Topicality. Offices and accommodations are often located close to various noise sources and must have sound insulation able to keep maximum noise level within an acceptable range. Poor sound insulation is the source of the negative effects of noise on human health and productivity, privacy breaches or information leakage and significantly reduces the cost of premises. As a consequence, there is a need to evaluate sound insulation efficiency during the design or reconstruction of premises.

In architectural design, the acoustic characteristics of the designed environment — a room or a building — are often represented by some numerical values or visually, for example, using heat maps. Auditory assessment of how a building fabric effects sound from existing noise sources will create an additional way to make an acoustic design of the premises more visual and opportunity to demonstrate the design results to users who are not acoustic experts. For this, we have to develop an airborne sound propagation auralization method and its software implementation. Auralization is a set of methods for rendering audible sound fields using numerical data obtained by simulation, measurement or synthesis. These methods are often used to create interactive environments for virtual reality applications, for example, in education and training programs.

The object of research is the process of software transformation of the audio signal.

The subject of the study is the methods and algorithms of software modeling of acoustic space.

The purpose of the research is to develop a method that models sound propagation within a specified set of rooms from sources to a listener and its software implementation.

Research methods used in this work include software engineering methods, computer modeling, and empirical methods.

The scientific novelty of the work is the development of an auralization method for airborne noise propagation in buildings, a method that is easy to implement and performance-wise effective.

The method can be used to process audio in real-time with the least possible loss of sound realism. To minimize the number of computations we take into account only direct noise propagation paths with no diffraction and reverberation, and sound-emitting partitions are considered as secondary point sources.

The practical value of the results obtained in the work is that the proposed method provides means for auditory assessment of the change in noise that propagates through the building fabric.

Approbation. The results of the research have been presented and discussed at the 12th scientific conference for graduate and postgraduate students “Applied mathematics and computing” PMK-2019.

Structure and content of the thesis. The master thesis consists of the introduction, five chapters, conclusions, and appendices.

The introduction presents the general description of the work and results of the assessment of the problem’s current state, substantiates the research topicality.

The first chapter provides an overview of general approaches and technical means used for auralization, along with examples of room acoustics simulation methods.

In the second chapter purpose and objectives of the study are formulated; analysis of mathematical models of the sound propagation in buildings and design techniques of digital filters for audio signals is performed; proposed auralization method for airborne noise propagation within a defined set of rooms.

The third chapter presents the results of requirements analysis for the software implementation of the method — standalone application built with the Unity engine. Its architecture and key algorithms are documented as well.

The fourth chapter provides testing methodology developed to ensure that given requirements to the software product are met, describes created sets of test data and environment models, gives an analysis of the obtained results and states further required improvements.

In the fifth chapter, we evaluated the commercial viability of the software implementation of the method by considering it as a main product for the startup. Analysis results are summarized in the business model and potential profitability calculations.

The conclusions are based on the analysis of the obtained results.

The appendices contain source code fragments of the software implementation of the method, a copy of the drawings and presentation slides.

The thesis is presented in 84 sheets, it has 3 appendices, 34 citations, 23 figures, and 7 tables.

Keywords: auralization, sound field modeling, acoustics, sound propagation, noise control, sound insulation, Unity.

РЕФЕРАТ

Актуальность. Офисные и жилые помещения часто расположены рядом с источниками шума различных типов и требуют звукоизоляции, способной снизить максимальный уровень шума в них до приемлемого. Неэффективная звукоизоляция становится источником негативного влияния шума на здоровье и продуктивность человека, нарушения приватности и утечки информации, а также снижает стоимость помещения. Как следствие, существует необходимость оценивать эффективность шумоизоляции на этапе проектирования или реконструкции помещений.

В архитектурном проектировании описание акустических характеристик спроектированного окружения – помещения или здания – представляется некоторыми числовыми величинами и визуально, например, с помощью тепловых карт. Возможность на слух оценить влияние конструктивных элементов окружения на звук от имеющихся источников шума придаст наглядности процессу проектирования акустики помещений и позволит демонстрировать результаты проектирования пользователям, не являющимся специалистами в области акустики. Для реализации такой возможности предлагается разработать способ аурализации передачи воздушного шума и его программную реализацию. Аурализацией называют совокупность способов воспроизведения звуковых полей на основе числовых данных, полученных путём моделирования, измерения или синтеза. Их часто применяют при создании интерактивных окружений в средствах виртуальной реальности, например, учебных и тренировочных программах.

Объектом исследования является процесс программного преобразования аудиосигнала.

Предметом исследования способы и алгоритмы программного моделирования акустического пространства.

Целью исследования является разработка способа, позволяющего воспроизвести характер изменения звука в процессе распространения в пределах множества комнат от источника к слушателю, а также создание программной реализации способа.

Методы исследования. В работе используются методы программной инженерии, математического моделирования и эмпирические методы.

Научная новизна состоит в разработке способа аурализации распространения воздушного шума в здании, преимущество которого в простоте реализации и высокой эффективности.

Разработанный способ может применяться для обработки аудиосигнала в реальном времени с минимально возможной потерей реалистичности звучания. Для уменьшения количества необходимых вычислений учитываются только прямые пути распространения шума без возникновения дифракции и реверберации, а перегородки, излучающие звук, считаются вторичными точечными источниками.

Практическая ценность полученных в работе результатов состоит в том, что предложенный способ предоставляет средства, позволяющие на слух оценить влияние на звук источника шума прохождение через конструктивные элементы строения.

Апробация работы. Основные положения и результаты работу докладывались и обсуждались на XII научной конференции магистрантов и аспирантов «Прикладная математика и компьютеринг» ПМК-2019.

Структура и объем работы. Магистерская диссертация состоит из введения, пяти глав, заключения и приложений.

Во введении дана общая характеристика работы, выполнена оценка современного состояния проблемы, обоснована актуальность направления исследований.

В первом разделе рассмотрены общие подходы и технические средства, используемые в аурализации, приводятся примеры методов, применяемые в симуляции акустики помещений.

Во втором разделе сформулирована научно-инновационная задача, проведён анализ математических моделей распространения звука в зданиях и рассмотрены подходы к созданию цифровых фильтров для обработки аудиосигнала, предложен способ аурализации распространения воздушного шума в пределах множества комнат здания от источника к слушателю.

В третьем разделе проведён анализ требований к программной реализации разработанного способа аурализации – прикладной программы, построенной с использованием программного движка Unity. Приведены архитектура и ключевые алгоритмы программного продукта.

В четвёртом разделе описана методика тестирования программного продукта для обеспечения соответствия сформулированным требованиям, создан набор тестовых данных и моделей окружения, выполнен анализ полученных результатов и определены необходимые усовершенствования.

В пятом разделе с целью оценить коммерческую жизнеспособность программной реализации разработанного способа рассмотрена возможность создания стартап-проекта на её основе. Построена бизнес-модель и приведены показатели потенциальной прибыльности.

В заключении проанализированы полученные результаты работы.

В приложениях приведены фрагменты исходного кода программной реализации способа, копии графических материалов и слайдов презентации.

Работа выполнена на 84 листах, содержит 3 приложения и ссылки на список использованных литературных источников из 34 наименований. В работе приведены 23 рисунка и 7 таблиц.

Ключевые слова: аурализация, моделирование звуковых полей, акустика, распространение звука, шумоизоляция, звукоизоляция, Unity.

ЗМІСТ

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ.....	4
ВСТУП.....	7
1. АНАЛІЗ ТЕОРЕТИЧНИХ ЗАСАД МОДЕЛЮВАННЯ АКУСТИЧНИХ ЯВИЩ.....	9
1.1. Теоретичні відомості про звук.....	9
1.2. Основні засади психоакустики.....	14
1.3. Класифікація та допустимі рівні шуму в будівлях.....	16
1.4. Ауралізація як засіб моделювання звукових полів.....	19
1.5. Застосування ауралізації в симуляції акустики приміщень.....	24
1.6. Висновки до розділу 1.....	26
2. СПОСІБ АУРАЛІЗАЦІЇ ЯКОСТІ ЗВУКОІЗОЛЯЦІЇ ПРИМІЩЕНЬ.....	27
2.1. Постановка науково-інноваційної задачі.....	27
2.2. Аналіз математичних моделей.....	27
2.3. Розроблення цифрових фільтрів.....	30
2.4. Формулювання способу ауралізації.....	41
2.5. Висновки до розділу 2.....	44
3. РОЗРОБЛЕННЯ ІННОВАЦІЙНОГО ПРОГРАМНОГО ПРОДУКТУ.....	46
3.1. Вимоги до програмного продукту.....	46
3.2. Вибір засобів розроблення.....	48
3.3. Архітектура програмного продукту.....	51
3.4. Алгоритм пошуку елементарних шляхів в мультиграфах.....	52
3.5. Висновки до розділу 3.....	55
4. ТЕСТУВАННЯ ІННОВАЦІЙНОГО ПРОГРАМНОГО ПРОДУКТУ ТА АНАЛІЗ РОЗРОБЛЕНОГО СПОСОБУ.....	57
4.1. Методика тестування.....	57
4.2. Створення тестових сценаріїв для ручного тестування.....	57
4.3. Створення тестових моделей приміщень.....	62
4.4. Створення тестового набору акустичних подій.....	64
4.5. Аналіз отриманих результатів.....	65

4.6. Висновки до розділу 4.....	65
5. СТАРТАП.....	66
5.1. Аналіз проблеми.....	66
5.2. Зацікавлені сторони.....	67
5.3. Комерційне рішення. Основні характеристики.....	70
5.4. Конкурентні переваги рішення.....	71
5.5. Клієнти. Сегменти ринку споживання.....	72
5.6. Унікальна ціннісна пропозиція.....	73
5.7. Доходи та витрати.....	73
5.8. Бізнес-модель.....	75
5.9. Висновки до розділу 5.....	77
ВИСНОВКИ.....	79
СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....	81
ДОДАТКИ.....	85

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

Амплітудно-частотна характеристика, АЧХ – функція залежності амплітуди вихідного сигналу від частоти вхідного сигналу сталої амплітуди.

Артефакт – дефект, шум у сигналі, зображенні, звукозаписі, причиною якого є систематичні перешкоди або особливості використаних технічних (як апаратних, так і програмних) засобів.

Ауралізація, auralization – сукупність способів відтворення звукових полів на основі числових даних, отриманих шляхом моделювання, вимірювання чи синтезу.

Гц – герц, одиниця виміру частоти.

дБ – децибел, одиниця виміру рівня звукового тиску.

дБА – рівень звукового тиску в децибелах, виміри якого проводилися з врахуванням особливостей сприйняття гучності звуку людиною.

ДБН – Державні Будівельні Норми.

ДСН – Державні Санітарні Норми.

Елементарний шлях – шлях в графі, який не проходить двічі через одну вершину.

Па – паскаль, одиниця вимірювання тиску.

Передатна функція – в загальному випадку – функція залежності вихідних значень системи від вхідних. В обробці сигналів мають на увазі залежність амплітуди вихідного сигналу від частоти синусоїдального сигналу з рівною амплітудою на вході.

Band-pass filter – смуговий фільтр.

Band-stop filter – режекторний фільтр.

BFS – див. Breadth First Search.

Breadth First Search – пошук в ширину.

Cross-Talk Cancellation – придушення перехресних завад

CTC – див. Cross-Talk Cancellation.

Cutoff frequency – частота зрізу.

Depth First Search – пошук в глибину.

DFS – див. Depth First Search.

Finite Impulse Response, FIR – скінченна імпульсна характеристика.

Head Related Transfer Function, HRTF – передатна функція, яка характеризує зміну звуку на шляху від точки в просторі до слухового апарату людини, викликані її анатомічними особливостями.

High-pass filter – фільтр високих частот.

Infinite Impulse Response, IIR – нескінченна імпульсна характеристика.

Interaural Level Difference, ILD – різниця рівнів звукового тиску для правого та лівого вуха людини, один із факторів бінаурального ефекту.

Interaural Time Difference, ITD – затримка приходу сигналу з однаковими фазами для правого та лівого вуха людини, один із факторів бінаурального ефекту.

Just-Noticeable Difference, JND – мінімальна різниця в інтенсивності впливу подразника, яка може бути розрізнена або сприйнята людиною.

Low-pass filter – фільтр низьких частот.

Passband – смуга пропускання.

SDK (Software Development Kit) – набір засобів розроблення, утиліт та документації, що дозволяють програмістам створювати прикладні програми за визначеною технологією або для певної платформи.

Sound Pressure Level – рівень звукового тиску.

SPL – див. Sound Pressure Level.

Stopband – смуга затримки.

Text-to-Speech – синтез мови на основі тексту.

TTS – Text-to-Speech.

ВСТУП

Офісні та житлові приміщення в більшості випадків розміщені поряд з джерелами шуму різних типів – сусідніми приміщеннями, ланками транспортної мережі, промисловими підприємствами, автостоянками, стадіонами та спортмайданчиками тощо. Відомо, що ступінь впливу шуму на людину залежить від його рівня, характеру та тривалості впливу, а також індивідуальних особливостей людини. Основний вплив шуму доводиться на органи слуху людини. Тривала дія шуму високої інтенсивності призводить до поступової втрати слуху – починаючи з високих частот і поступовим поширенням на нижчі.

Окрім безпосереднього впливу на орган слуху, шум може впливати на процеси вищої нервової діяльності, призводячи до підвищеної втомлюваності, зниженої працездатності, роздратованості, апатії, послаблення уваги та пам'яті, тощо. Таким чином, шум сприяє зниженню продуктивності та виникненню нещасних випадків.

Несприятлива дія навіть незначних за рівнем шумів сильніше проявляється у години відпочинку. Тому, згідно з законом України “Про забезпечення санітарного та епідемічного благополуччя населення”, у житлових масивах діє заборона порушення тиші з двадцять другої до восьмої години, а шум при здійсненні будь-якої діяльності не повинен перевищувати рівнів, встановлених санітарними нормами для відповідного часу доби [1]. Крім того, “проведення ремонтних робіт, що супроводжуються шумом, забороняється у робочі дні з двадцять першої до восьмої години, а у святкові та неробочі дні – цілодобово”. Окремим винятком є лише згода мешканців усіх прилеглих квартир.

ДБН В.1.1-31:2013 та ДСН 3.3.6.037-99 є головними документами, що встановлюють норми допустимих рівнів шуму на робочих місцях, територіях та приміщеннях будинків різного призначення, правила проєктування засобів захисту від шуму та методику оцінки шумового режиму [2, 3].

З огляду на вищезначену важливість забезпечення прийняттого рівня шуму у приміщеннях, необхідність оцінювати ефективність шумоізоляції на етапі їх проєктування або реконструкції завжди була актуальною.

В архітектурному проєктуванні результати моделювання проникнення шуму в приміщення подаються певними числовими величинами та візуально, наприклад, за допомогою теплокарт. Таке подання є зручним та зрозумілим в більшості випадків лише спеціалістам. Наявність можливості на слух оцінити вплив елементів оточення на звук від наявних джерел підвищить наочність процесу проєктування та дозволить демонструвати результати користувачам-неспеціалістам. Для реалізації такої можливості пропонується розробити спосіб ауралізації поширення повітряного шуму та його програмну реалізацію.

1. АНАЛІЗ ТЕОРЕТИЧНИХ ЗАСАД МОДЕЛЮВАННЯ АКУСТИЧНИХ ЯВИЩ

1.1. Теоретичні відомості про звук

Звук – явище поширення хвиль у пружному середовищі (твердому, рідкому або газоподібному). Частіше за все, під звуком розуміють коливання, які сприймаються органами чуття людини та тварин.

Звукові хвилі створюються коливаннями джерела звуку, наприклад, камертона. Вібруючи, він спричиняє періодичну зміну тиску прилеглого середовища в одному місці. Оскільки атоми або молекули середовища пов'язані пружними зв'язками, тиск поступово передається на сусідні частинки, в результаті чого область підвищеного тиску переміщується у просторі. Області підвищеного тиску чергуються з областями зниженого, утворюючи хвилю. При цьому поширення хвиль не супроводжується перенесенням частинок середовища, вони лише коливаються відносно власного положення рівноваги. Крім того, коливання тиску відбуваються швидше, ніж поширення локальних змін температури на інші ділянки середовища, тобто процес є адіабатним [4].

Область середовища, в якій розповсюджуються звукові хвилі, називають звуковим полем.

Коливальний рух частинок середовища, що спостерігається у площині паралельній до напрямку його поширення, утворює повздовжні хвилі. Однак у твердих середовищах коливання можуть відбуватися також у площині, що перпендикулярна до напрямку поширення, такі хвилі називаються поперечними.

На рис. 1.1 проілюстрована зміна тиску з часом у певній точці середовища після виникнення звуку. Такі відхилення відносно статичного тиску називають миттєвим звуковим тиском p_i . Середньоквадратичне значення миттєвого тиску протягом деякого проміжку часу T називають діючим звуковим тиском:

$$p = \sqrt{\frac{1}{T} \int_0^T p_i^2(t) dt} . \quad (1.1)$$

В акустиці замість діючого звукового тиску у паскалях частіше використовують рівень звукового тиску (SPL), що є логарифмом відношення величини звукового тиску до деякого еталонного значення і вимірюється в децибелах. Для обчислення рівня звукового тиску використовують наступну формулу:

$$L_p = 10 \log_{10} \left(\frac{p^2}{p_0^2} \right) = 20 \log_{10} \left(\frac{p}{p_0} \right), \quad (1.2)$$

де p_0 – еталонне значення звукового тиску.

За еталонне значення у повітрі використовують $p_0 = 2 \cdot 10^{-5}$ Па, яке розглядають як поріг чутності звуку частотою 1 кГц для людини. При цьому для інших середовищ (наприклад, у підводній акустиці) за еталонне значення часто обирають величину в $1 \cdot 10^{-6}$ Па. Еталонні значення наведено у стандарті ANSI S1.1-2013.

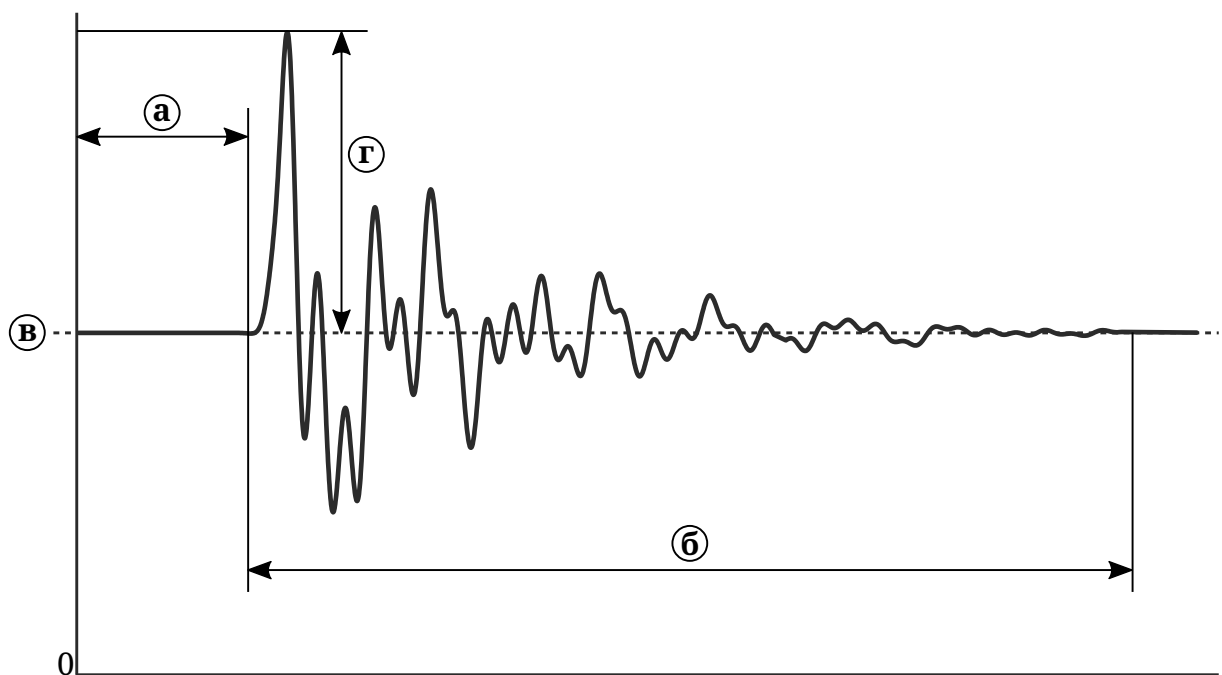


Рис. 1.1. Графік звукового тиску:

а) тиша; б) чутний звук; в) статичний тиск; г) миттєвий звуковий тиск.

Швидкість поширення звуку найвища у твердих середовищах, а найнижча – у газах, що пов'язано зі зменшенням жорсткості середовища. В загальному випадку маємо наступні формули обчислення швидкості звуку в ідеальному газі (1.3), рідині (1.4) та твердих тілах – окремо для поздовжніх (1.5) та поперечних хвиль (1.6):

$$c_g = \sqrt{\gamma \frac{p}{\rho}}, \quad (1.3)$$

$$c_l = \sqrt{\frac{K}{\rho}}, \quad (1.4)$$

$$c_{s,p} = \sqrt{\frac{K + \frac{4}{3}G}{\rho}}, \quad (1.5)$$

$$c_{s,s} = \sqrt{\frac{G}{\rho}}, \quad (1.6)$$

де γ – показник адіабати;

ρ – густина;

p – тиск;

K – модуль всебічного стиску;

G – модуль зсуву.

При цьому для наближеного розрахунку залежності швидкості звуку у повітрі від температури використовують формулу (1.7), побудовану лінеаризацією розкладання в ряд Тейлора більш точної формули (1.8) [5]:

$$c_a = 331,501 + 0,5857 \theta, \quad (1.7)$$

$$c_a = 343,215 \sqrt{\frac{273,15 + \theta}{293,15}}, \quad (1.8)$$

де θ – температура у градусах Цельсія.

Поширення звукових хвиль у середовищі можна розглядати як перенесення випромінюваної джерелом енергії. Джерело звуку при цьому характеризується значенням звукової потужності – загальної кількості звукової енергії, випромінюваної ним у всіх напрямках за одиницю часу. Звуковою енергією називається енергія коливань частинок середовища у

певному його об'ємі, яку можна подати як суму густин потенціальної та кінетичної енергій проінтегрованих за цим об'ємом (1.9). Звукову потужність, що проходить через певну поверхню за одиницю часу у напрямку перпендикулярному до даної поверхні називають звуковим потоком. Для плоских хвиль його значення обчислюють за формулою (1.10). При цьому відношення величини звукового потоку на одиницю площі визначає величину інтенсивності звуку (1.11). Для сферичної хвилі до неї застосовний закон обернених квадратів – її величина обернено пропорційна квадрату відстані до джерела звуку (1.12).

$$W = \int_V \frac{p^2}{2\rho_0 c^2} dV + \int_V \frac{\rho v^2}{2} dV, \quad (1.9)$$

$$P = \frac{S p^2}{\rho_0 c}, \quad (1.10)$$

$$I = \frac{P}{S}, \quad (1.11)$$

$$I(r) = \frac{P}{4\pi r^2} \propto \frac{1}{r^2}, \quad (1.12)$$

де p – ефективний звуковий тиск;

ρ_0 – густина середовища в стані спокою;

ρ – локальна густина середовища;

v – швидкість частинок середовища;

V – об'єм;

S – площа проекції поверхні на площину хвилі;

r – відстань до джерела звуку.

Число коливань за одиницю часу називають частотою f , одиницею виміру якої є герц (Гц). Звук з частотою нижче 20 Гц називають інфразвуком, в межах 20 кГц–1 ГГц – ультразвуком. Швидкість розповсюдження хвиль у просторі пов'язана із частотою коливань співвідношенням (1.14):

$$f = \frac{1}{T}, \quad (1.13)$$

$$c = f \lambda = \frac{\lambda}{T}, \quad (1.14)$$

де λ – довжина хвилі, тобто її просторовий період;

T – період коливань у часі.

Для спрощення вимірів та обчислень в акустиці весь діапазон частот ділять на групи частот, які називають смугами частот. Найбільш розповсюджений поділ на октавні смуги, в яких верхня гранична частота f_u вдвічі вища за нижню f_l , а також третиннооктавні смуги. Середина смуги m визначається як середнє геометричне нижньої та верхньої границі та називається центральною частотою f_m . Співвідношення між частотами для октавних та третиннооктавних смуг наведено у формулах (1.15) та (1.16), відповідно:

$$\begin{aligned} f_u &= 2f_l, \\ \Delta f &= f_u - f_l = f_l, \\ f_m &= \sqrt{f_u \cdot f_l} = \sqrt{2}f_l, \end{aligned} \quad (1.15)$$

$$\begin{aligned} f_{m+1} &= 2f_m, \\ f_u &= \sqrt[3]{2}f_l, \\ \Delta f &= f_u - f_l = (\sqrt[3]{2} - 1)f_l, \\ f_m &= \sqrt[3]{f_u \cdot f_l}, \\ f_{m+1} &= \sqrt[3]{2}f_m. \end{aligned} \quad (1.16)$$

Номінальні центральні октавні частоти є стандартизованими: 31,5, 63, 125, 250, 500, 1000, 2000, 4000, 8000, 16000 Гц [6, 7]. За необхідності у більш детальному аналізі використовуються вузьчі частотні смуги, наприклад шириною 1/6 або 1/12 октави.

1.2. Основні засади психоакустики

Психоакустика – наукова дисципліна, що вивчає психологічні та фізіологічні особливості сприйняття людиною звуку.

Слухова система середньостатистичної людини здатна сприймати звук з частотного діапазону між 20 Гц та 20 кГц, а найбільш чутлива – до частот між 2 та 5 кГц.

Суб'єктивною характеристикою сприйняття людиною сили звукового тиску є гучність. Одиницею виміру гучності за абсолютною шкалою є сон, яка відповідає гучності синусоїдального тону частотою 1 кГц з рівнем звукового тиску 40 дБ.

Гучність звуку залежить не лише від сили звукового тиску, а й від частотної складової та тривалості звуку. Один і той же звук за тривалості меншої часового вікна інтегрування слухової системи людини, яке становить приблизно 600..1000 мс, сприйматиметься тихішим зі зменшенням його тривалості.

Залежність гучності від рівня звукового тиску та частоти для чистих синусоїдальних тонів експериментальним шляхом першими визначили Флетчер та Мансон у 1933 році. Результатом їх досліджень стали так звані “криві Флетчера-Мансона” або “криві рівних гучностей”. Це сімейство кривих було пізніше уточнене та стандартизоване [8], їх наведено на рис. 1.2. Числа поряд з кривими вказують на рівень гучності, найнижча крива відповідає порогу чутності – 3 фони. Фон – одиниця виміру рівня гучності, яка відповідає рівню звукового тиску синусоїдального тону частотою 1 кГц, що відчувається однаково гучним відносно звуку, який оцінюють. Криві рівних гучностей дозволяють визначити рівень гучності чистого синусоїдального тону за його рівнем звукового тиску та частотою. Наприклад, тон з рівнем звукового тиску 50 дБ та частотою 1 кГц матиме рівень гучності 50 фон. При цьому для частоти 250 Гц його рівень гучності становитиме лише 40 фон, а при 40 Гц – цей звук майже не чуто.

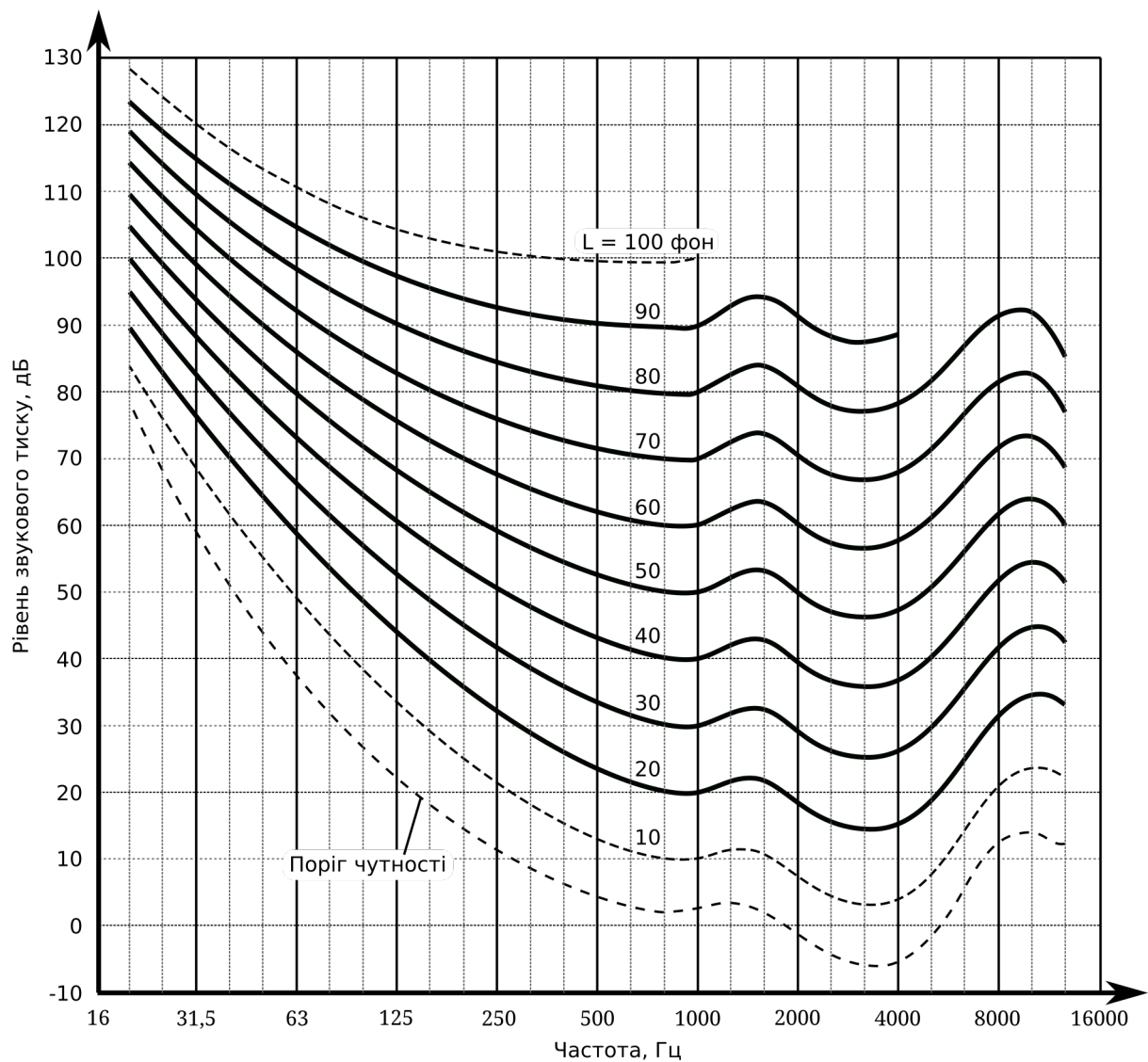


Рис. 1.2. Криві рівних гучностей відповідно до ISO 226:2003.

Сприйняття зміни гучності звуку людиною наближено підкоряється емпіричному закону Вебера-Фехнера. Відповідно до цього психофізичного закону, інтенсивність відчуття людини має логарифмічну залежність від інтенсивності подразника:

$$p = k \ln \frac{S}{S_0}, \quad (1.17)$$

де p – міра відчуття, сприйнятого людиною;

k – константа, яка залежить від модальності відчуття;

S – інтенсивність подразника;

S_0 – нижня гранична інтенсивність подразника.

Таким чином, рівень звукового тиску в децибелах більш точно відповідає сприйняттю гучності звуку людиною у порівнянні з величиною звукового тиску у паскалях, а 1 дБ розглядають як мінімальну відчутну різницю (JND). Іншим наслідком логарифмічної залежності є і те, що людина може почути певний звук на фоні іншого лише тоді, коли його амплітуда перевищує амплітуду фонового звуку не на деяку певну абсолютну величину, а на певний множник, який залежить від частоти.

Здатність людини визначати розміщення джерел звуку залежить від числа фізичних факторів, серед яких виділяють наступні:

- часова затримка приходу сигналів з однаковими фазами – ITD;
- різниця рівнів звукового тиску для правого та лівого вуха через дифракцію звукової хвилі навколо голови – ILD;
- різниця спектрального складу внаслідок відбиття, поглинання та дифракції, причиною яких є вушні раковини, голова та торс;
- затухання амплітуди, втрата високих частот та співвідношення між прямим сигналом до ревербераційного відлуння, які впливають на визначення відстані до джерела.

Принципи виникнення ITD та ILD проілюстровано на рис. 1.3.

Крім того, на локалізацію звуку впливають рухи голови.

1.3. Класифікація та допустимі рівні шуму в будівлях

Шумом називають будь-який небажаний звук, що сприймається негативно, заважає слуховому сприйняттю корисної інформації, завдає шкоди здоров'ю або знижує працездатність людини. Відповідно до ДБН В.1.1-31:2013 встановлено наступні допустимі рівні шуму [2]:

- у житлових приміщеннях 30 дБА в нічний час доби та 40 дБА – у денний;
- в офісних приміщеннях – 50 дБА;
- для аудиторій, навчальних кабінетів та кімнат викладачів навчальних закладів, читальних залів бібліотек – 40 дБА;

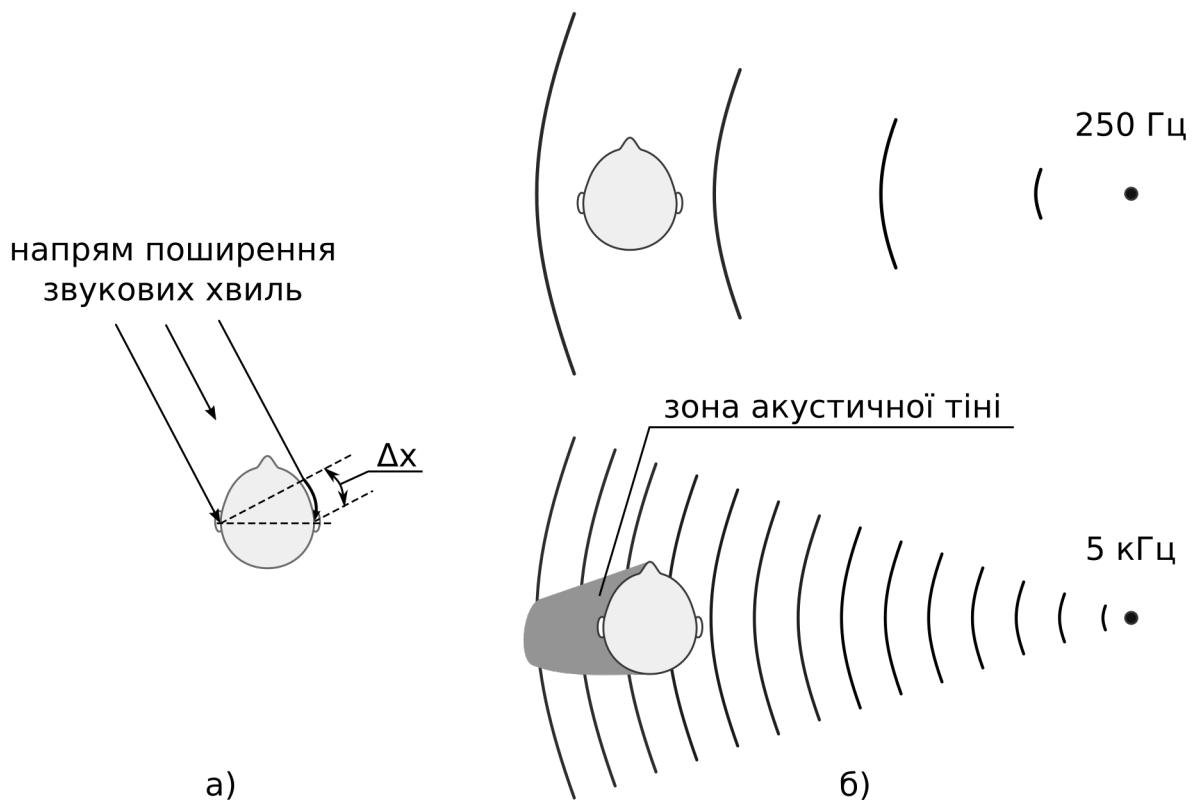


Рис. 1.3. Виникнення фізичних факторів, які впливають на локалізацію звуку людиною: а) ITD, де Δx – різниця ходу; б) ILD.

- в кабінетах лікарів – 35 дБА;
- в палатах лікарень та санаторіїв – 35 дБА в денний час доби та 25 дБА – в нічний;
- в операційних приміщеннях лікарень – 30 дБА.

За тривалістю шум поділяють на постійний та непостійний. Рівень звуку постійного шуму змінюється в часі не більше ніж на 5 дБА. До непостійного шуму належить мінливий, імпульсний та переривчастий. У мінливого шуму рівень звуку безперервно змінюється у часі. Імпульсний шум складається з одного, декількох або періодичних сигналів з тривалістю до однієї секунди. Рівень звуку переривчастого шуму змінюється ступінчасто з тривалістю сталого рівня щонайменше однієї секунди.

За частотним складом виділяють широкосмугові та тональні шуми. Широкосмугові шуми мають безперервний спектр шириною більше однієї

октави. В спектрі тональних шумів можливо виділити дискретні тони – рівень шуму в третиннооктавній смузі частот має перевищувати його рівні в сусідніх смугах щонайменше на 10 дБ.

В залежності від походження виділяють антропогенні (ті, що виникли в результаті діяльності людини – від промислових, будівельних та сільськогосподарських об'єктів, транспорту, концертів, спортивних подій тощо) та природні шуми.

В поширенні шуму у будівлях між двома приміщеннями виділяють два види шляхів: прямі та непрямі (другорядні), їх приклад проілюстровано на рис. 1.4. Під прямими шляхами розуміють передачу звуку з одного приміщення в інше через спільні конструктивні елементи-перегородки.

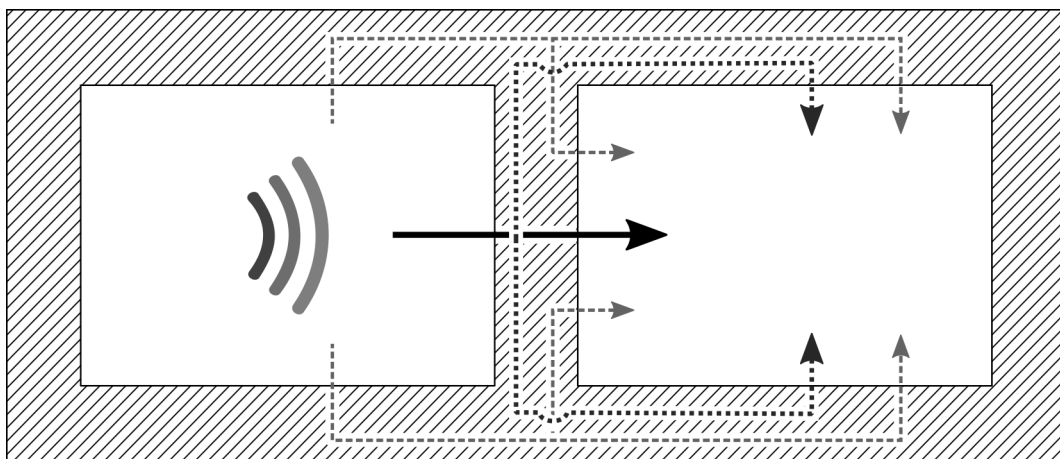


Рис. 1.4. Приклад шляхів поширення шуму між двома приміщеннями. Суцільною лінією позначено прямий шлях, переривчастою – непрямий

В залежності від середовища поширення виділяють два пов'язаних між собою типи шуму: повітряний та структурний. Повітряним є шум, що випромінюється та поширюється у повітрі, наприклад, розмови людей, гавкіт собак тощо. Структурний шум представлений поширенням вібрацій у будівельних конструкціях. Повітряний шум викликає появу вібрацій у конструктивних елементах будівлі при падінні на площину їх поверхні. У свою чергу, ці вібрації можуть викликати коливання поверхонь

структурних елементів та породжувати повітряний шум. На рис. 1.5 наведено приклади поширення цих типів шуму в будівлі. Іноді виділяють третій тип – ударний, який є наслідком механічного впливу твердих тіл на структурні елементи будівлі, наприклад, звуки кроків, удари в стіну, падіння предметів тощо.

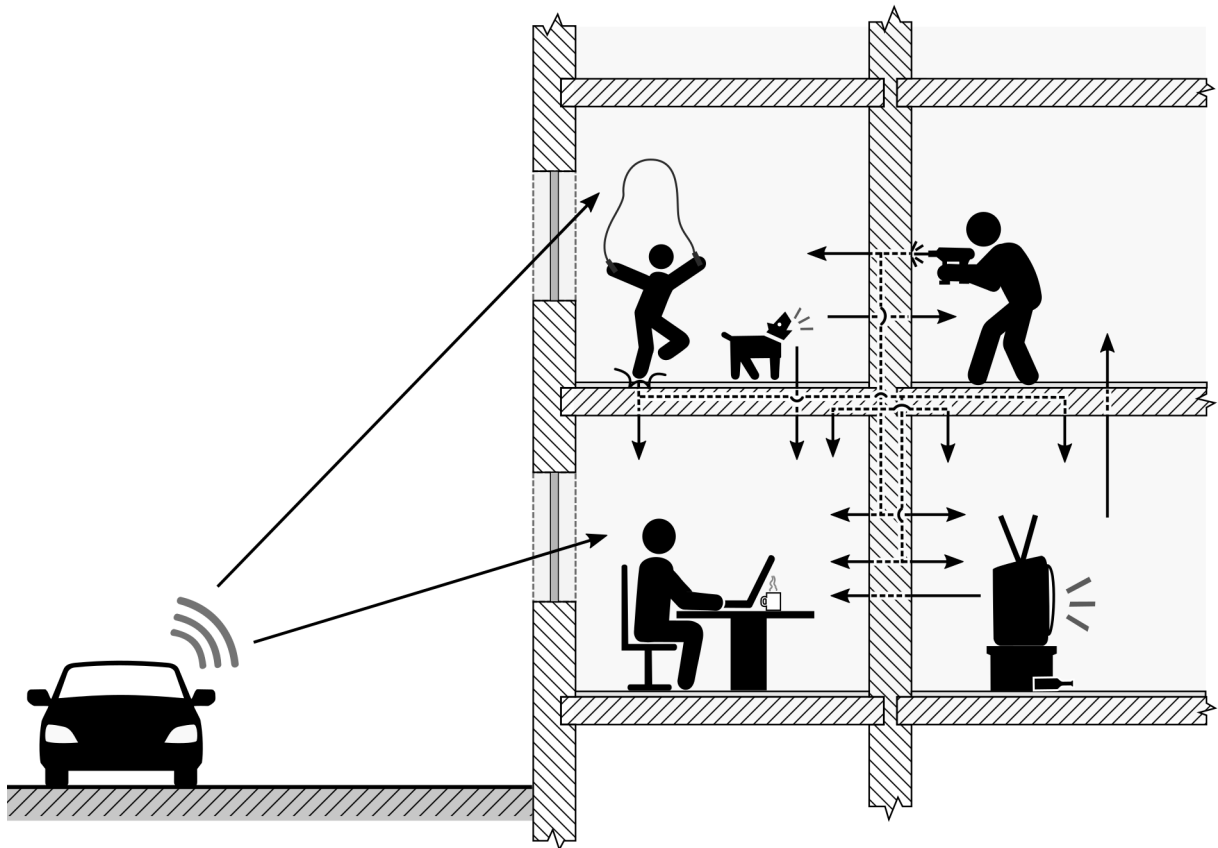


Рис. 1.5. Приклад типових джерел шуму та його поширення в будівлі.

Суцільною лінією позначено повітряний шум, переривчастою –
структурний

1.4. Ауралізація як засіб моделювання звукових полів

Термін “ауралізація” вперше з’явився у 1993 році, його ввели для опису відтворення звукових полів по аналогії до терміну “візуалізація” у статті за авторством Кляйнера, Даленбека та Свенссона [9]. Вони визначили цей термін наступним чином: “Ауралізація – це процес відтворення чутного звукового поля джерела звуку у просторі з використанням фізичного або математичного моделювання, таким чином

імітувати бінауральний ефект прослуховування у певному місці модельованого оточення”. Більш загальне визначення дає Форлендер: “Ауралізація – спосіб створення звукових файлів на основі числових даних, отриманих шляхом моделювання, вимірювання чи синтезу” [5]. Іншим розповсюдженим синонімічним терміном є “віртуальна акустика” [10].

Одне з найочевидніших застосувань ауралізації – створення інтерактивних оточень у засобах віртуальної реальності, наприклад, в навчальних та тренувальних програмах. В архітектурному проєктуванні використання ауралізації стає чудовим доповненням опису акустичних властивостей спроектованого оточення (наприклад, кімнати) певними числовими величинами та візуального їх подання (наприклад, показом теплокарти). В цьому випадку ауралізація створює можливість на слух оцінити вплив елементів оточення на звук від наявних джерел, а це додає як наочності процесу проєктування акустики приміщень, так і дозволяє продемонструвати результати проєктування неспеціалістам.

В ауралізації можливо виділити моделювання трьох пов’язаних між собою складових: джерел звуку, середовища та слухача [5, 10]. З точки зору обробки сигналів результат модельованих процесів можна розглядати як згортку імпульсної характеристики середовища та необробленого аудіосигналу. Зв’язки між модельованими складовими та відповідне їх подання в обробці сигналів наведено на рис. 1.6.

Джерелам звуку відповідають методи, що моделюють породження звуку в середовищі. Найпростіший із них – використання попередньо записаного (частіше за все – у безлунній камері) необробленого звуку. Складніші методи генерують звук із використанням певних алгоритмів – синтез мовлення (TTS), музики, процедурна генерація шуму, в тому числі – з використанням фізичної симуляції взаємодії об’єктів.

В найпростіших випадках достатньо розглядати ненаправлені точкові джерела звуку, проте виникає необхідність й у більш точному моделюванні.

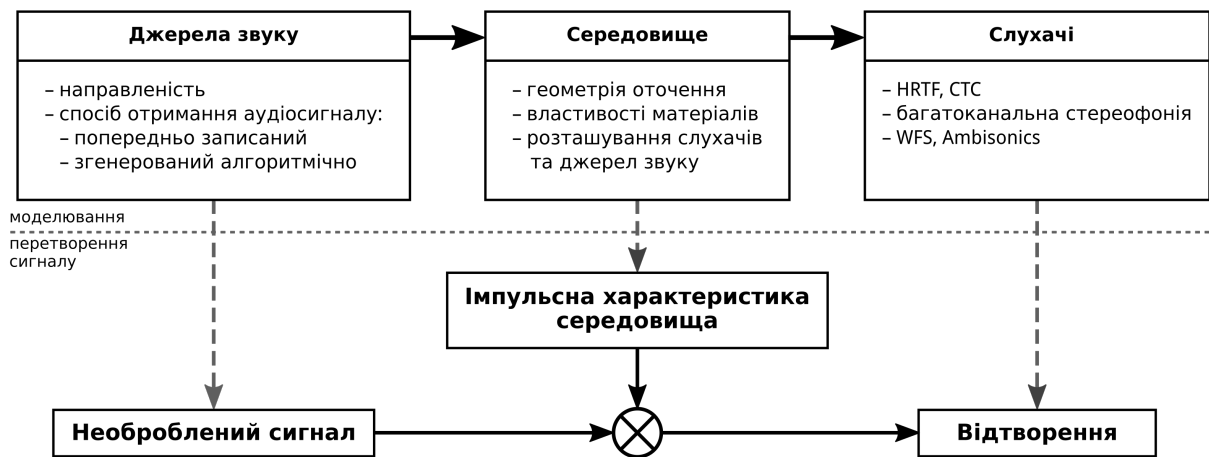


Рис. 1.6. Складові процесу ауралізації та відповідні засоби в моделюванні та перетворенні сигналу

Зазвичай джерела звуку (наприклад, людина, що говорить, гучномовці та музичні інструменти) випромінюють максимальну кількість енергії у фронтальній напівсфері, при віддаленні від осі якої сила звуку спадає разом з послабленням високих частот. Крім того, більшість музичних інструментів мають ще й частотно-залежну направленість, тіло музиканта спричиняє розсіювання, а звук може виникати в різних частинах інструмента в залежності від ноти, яку грають. Оскільки в таких випадках використання математичних методів є складною задачею, тому для спрощення моделювання використовують результати вимірювань напрямленості.

При моделюванні напрямленості в реальному часі застосовують два основних підходи: фільтрація за напрямом та використання множини елементарних джерел. В першому випадку вихід монофонічного джерела звуку подається на вхід цифрового фільтра разом з азимутальним (а за необхідності – і з полярним) кутом. У випадку, коли зміна напрямленості та сили випромінювання розподілена у просторі, використання точкового джерела не може бути достатнім для реалістичного відтворення. Наприклад, лінійне джерело або джерело складнішої форми можна апроксимувати набором точкових джерел.

Моделювання розповсюдження звуку у середовищі від джерел до слухачів ставить за мету відтворення імпульсної характеристики середовища, тобто характеру змін звуку, що виникають через явища відбиття, дифракцію, затухання та поглинання. Такі моделі застосовують для симуляції акустики приміщення (відтворення реверберації та чутності), поширення звуку ззовні (або навпаки – зсередини) або на відкритому просторі. Вхідними даними для них стають властивості матеріалів та геометрія оточення (частіше за все – максимально спрощена), розміщення слухачів та джерел в ньому. Такий набір даних називають сценою. В динамічному віртуальному оточенні джерела звуку, слухачі та об'єкти оточення можуть змінювати своє положення та орієнтацію в просторі, що призводить до необхідності повторно обчислювати імпульсну характеристику, а також виконувати інтерполяцію між старою та новою імпульсною характеристикою для уникнення появи чутних артефактів. В деяких випадках її обчислення можливо розділити на два етапи: автономний та інтерактивний.

Процес моделювання слухача виконує перетворення аудіосигналу з врахуванням особливостей локалізації звуку людиною, впливу її анатомії на сприйнятий звук та обладнання, яке використовується для відтворення звуку. Серед способів відтворення просторового звуку можна виділити два найбільш розповсюджені підходи: бінауральна стереофонія та багатоканальний звук.

Бінауральна стереофонія орієнтована на створення об'ємного звучання при прослуховуванні у навушниках або через гучномовці. При використанні навушників такі системи часто доповнюють засобами відстеження голови слухача, оскільки її рухи впливають на надійність локалізації звуку.

Для синтезу бінаурального сигналу до початкового (монофонічного) аудіосигналу застосовують HRTF – передавальну функцію, що характеризує зміну сприйнятого людиною звуку на шляху до її слухового

апарату з певної точки простору. Такий метод відтворює виникнення факторів локалізації звуку, наведених в підрозділі 1.2.

В загальному випадку HRTF є комплекснозначною функцією від частоти та сферичних координат.

HRTF є індивідуальною для кожної людини внаслідок залежності від анатомічних особливостей (розміру голови, форми вушної раковини) [11]. HRTF отримують вимірюванням різниці рівнів звукового тиску для кожного з вух в залежності від напрямку з використанням манекена (“штучної голови”) або внутрішньовушних мікрофонів у вухах людини. На сьогодні існують набори даних вимірів HRTF, в тому числі – з відкритим доступом [12].

На відміну від навушників, відтворення через гучномовці має недостатнє розмежування каналів правого та лівого вуха, що значно знижує точність локалізації. Для компенсації цього ефекту використовують методи придушення перехресних завад (СТС). Іншим недоліком відтворення через гучномовці є те, що стереофонічний ефект наявний лише в невеликій області, яке називають “sweet spot” [13].

Розширенням бінауральної стереофонії стали системи багатоканального відтворення звуку, наприклад, у типовій восьмиканальній конфігурації 7.1 surround sound. До неї входять два тилових канали, два бокових, три фронтальних та канал низькочастотних ефектів [14]. Багатоканальні системи також не повністю позбавлені від проблеми наявності “sweet spot”.

Альтернативою багатоканальній та бінауральній стереофонії є системи синтезу звукового поля. Такі системи намагаються відтворити fronti хвиль оригінального звукового поля у просторі, в якому знаходиться слухач. За аналогією до оптичної голографії ці технології отримали назву “голофонія”. Вони базуються на принципі Гюйгенса, відповідно до якого, будь-яка точка простору, якої досягла хвиля, може розглядатися як джерело вторинних сферичних хвиль. Головна перевага

такого підходу – здатність забезпечувати можливість надійної локалізації уявних джерел. Wave Field Synthesis є технологією, що реалізує синтез звукових полів, основний її недолік – висока вартість.

1.5. Застосування ауралізації в симуляції акустики приміщень

Більшість методів моделювання акустики приміщень розглядають розповсюдження звуку у вигляді променів. Метод трасування променів (ray tracing) почали застосовувати ще в 60-х роках минулого століття в тривимірній графіці для отримання реалістичних зображень, згодом його стали застосовувати в обробці звуку. В загальному випадку, розповсюдження звуку дійсно аналогічне променям світла, що виходять з джерела, проходять крізь повітря, відбиваються від поверхонь і досягають отримувача. Однак швидкість поширення та довжина хвиль у світла і звуку відрізняється. Як наслідок, для людини світло поширюється миттєво, а значно менша швидкість звуку (340 м/с у повітрі) є причиною таких явищ як луна та реверберація. Довжина хвилі світла (400-740 нм) надзвичайно мала і її слід брати до уваги лише в окремих випадках. Водночас довжина звукових хвиль, що сприймаються людиною, може досягати розмірів фізичних об'єктів (від 0,017 до 17 м). Перешкоди на шлях звуку призводять до значного затухання високочастотної складової, при цьому низькі частоти можуть проходити перешкоду без помітного згасання. Наприклад, ми можемо чути людину, яка говорить за рогом і є для нас поза межами прямої видимості, проте звук буде приглушеним. Окрім цього, ми змушені враховувати явище дифракції, наслідком якого є потрапляння звукових хвиль в область геометричної тіні.

Інші два променевих методи моделювання – метод уявних джерел (image-source method) та метод трасування пучків променів (beam tracing). Метод трасування пучків є розширенням методу трасування променів в тому сенсі, що променям надають об'єм. Метод уявних при віддзеркаленні “променя” звуку створює вторинне джерело “позаду”

поверхні віддзеркалення. Це джерело буде розташоване на прямій, що перпендикулярна до поверхні, та на тій же відстані до неї, що й оригінальне джерело. Якщо нове джерело є віддзеркаленням від однієї поверхні, воно називається віддзеркаленням першого порядку. Промінь, який віддзеркалюється від декількох поверхонь, буде представленим уявним джерелом вищих порядків. Всі джерела, і оригінал, і уявні, випромінюють один і той же сигнал одночасно, а отримана імпульсна характеристика (тобто, рівень звуку в часі) формується додаванням сигналів від кожного джерела, із врахуванням затримки та затухання відповідно до відстані між джерелом та отримувачем. Частотна характеристика сигналу кожного уявного джерела буде додатково змінена відповідно до характеристики поверхні віддзеркалення. На практиці, при віддзеркаленні зберігається не вся енергія звуку, частина буде розсіяна випадковим чином в інших напрямках, і моделювання методом уявних джерел не враховує це явище. Кутруф показав, що лише ранні віддзеркалення є здебільшого дзеркальними, після кількох віддзеркалень основна частина звукової енергії розсіюється [4]. Тому пропонується використовувати гібридні методи, початкові віддзеркалення обчислюються як уявні джерела, а пізні (розсіяні) – з допомогою інших моделей, наприклад, трасування променів.

Окрім променевих методів, дослідники застосовують хвильові методи, наприклад метод скінчених елементів (finite-element, FEM), метод граничних елементів (boundary element, BEM) та метод скінченої різниці у часі та просторі (finite-difference-time-domain, FDTD). Однак через значну обчислювальну складність такі методи не є практичними в застосуванні до всього діапазону частот, що сприймає людина, і не підходять для обчислень у реальному часі.

1.6. Висновки до розділу 1

В рамках даного розділу розглядалися загальні підходи та технічні засоби, що використовуються в моделюванні звукових полів – ауралізації. Наводяться приклади методів ауралізації, які застосовуються для симуляції акустики приміщень.

Також розглянуто теоретичні відомості про фізичні характеристики звукових хвиль та особливості їх сприйняття людиною, які враховуються при ауралізації. Оскільки робота присвячена моделюванню поширення шуму, додатково наводиться характеристики типів шуму та його допустимі рівні в будівлях відповідно до ДБН В.1.1-31:2013.

2. СПОСІБ АУРАЛІЗАЦІЇ ЯКОСТІ ЗВУКОІЗОЛЯЦІЇ ПРИМІЩЕНЬ

2.1. Постановка науково-інноваційної задачі

Задача полягає у розробленні інноваційного програмного продукту, який реалізує спосіб ауралізації поширення повітряного шуму в межах множини кімнат від джерел до слухача.

Розв'язання даної задачі вимагає виконання наступних кроків:

- проведення аналізу математичних моделей поширення звуку в будівлях;
- розгляду підходів до створення відповідних цифрових фільтрів для оброблення аудіосигналу;
- формулювання способу ауралізації.

2.2. Аналіз математичних моделей

Між приміщеннями будівлі основна частина звукової енергії передається прямими шляхами, тобто через суміжні конструктивні елементи, і незначна її частина передається другорядними шляхами.

При проходженні шуму через перегородку, енергія звукових хвиль, що падають на перегородку, частково відбивається від неї та поглинається в товщі матеріалу, це проілюстровано на рис. 2.1. Ступінь поглинання залежить від маси перегородки, а відбиття – від співвідношень імпедансів середовища та перегородки.

Основною характеристикою конструктивного елемента-перегородки, яка цікавить нас в моделюванні, є коефіцієнт звукопроникності:

$$\tau = \frac{I_t}{I_0} = 1 - \frac{I_a + I_r}{I_0}, \quad (2.1)$$

де I_0 – інтенсивність падаючих хвиль;

I_a – частина інтенсивності, що була поглинута;

I_r – інтенсивність відбитого звуку;

I_t – інтенсивність звуку, що пройшов через перегородку.

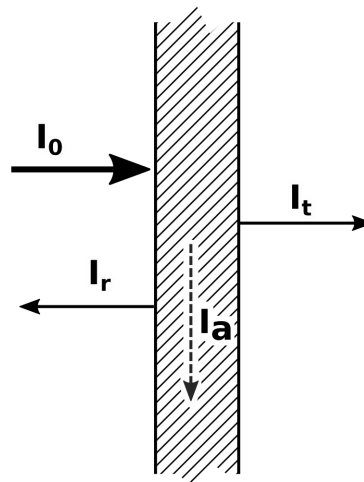


Рис. 2.1. Поширення звуку через перегородку

При падінні плоских хвиль на одношарову перегородку товщиною h під кутом θ_0 до нормалі коефіцієнт її звукопроникності визначається формулою (2.2). В даній моделі нескінченна перегородка оточена з боків рідким або газоподібним середовищем та поширення звуку в товщі перегородки відбувається лише у вигляді поздовжніх хвиль. Кут їх поширення θ_0 до напрямку протилежного нормалі визначається за законом Снеліуса (2.3).

$$\tau = \frac{1}{\cos^2(k h \cos \theta) + \frac{1}{4} \left(\frac{z_0}{z} + \frac{z}{z_0} \right) \sin^2(k h \cos \theta)}, \quad (2.2)$$

$$\cos \theta = \sqrt{1 - \left(\frac{c}{c_0} \sin \theta_0 \right)^2}, \quad (2.3)$$

де $k = \frac{2\pi}{\lambda} = \frac{\omega}{c}$ – хвильове число;

$z_0 = \frac{\rho_0 c_0}{\cos \theta_0}$ та $z = \frac{\rho c}{\cos \theta}$ – імпеданси середовища та перегородки;

c_0 та c – швидкості звуку у середовищі та перегородці.

Коефіцієнт звукопроникності приймає максимальне значення $\tau = 1$, тобто звукоізоляція відсутня, коли довжина цілого числа напівхвиль, спроектована на нормаль, дорівнює товщині перегородки:

$$k h \cos \theta = \frac{2\pi h}{\lambda} \cos \theta = \pi n, \quad (2.4)$$

$$\frac{\lambda}{2} n = h \cos \theta, \quad (2.5)$$

де $n \in \mathbb{Z}$.

Для перегородок з отворами відносно великих розмірів (вікна, двері), сумарний коефіцієнт звукопроникності визначається виразом:

$$\tau = \frac{\tau_p S_p + \tau_h S_h}{S_p + S_h}, \quad (2.6)$$

де τ_p – звукопроникність перегородки;

τ_h – звукопроникність отвору;

S_p – площа отвору;

S_p – площа перегородки, не враховуючи отвір.

В ньютонівських рідинах, таких як повітря та вода, відповідно до закону Стокса, амплітуда плоскої хвилі експоненційно спадає зі збільшенням відстані d :

$$A(d) = A_0 e^{-\alpha d}, \quad (2.7)$$

$$\alpha = \frac{2 \mu \omega^2}{3 \rho c^3}, \quad (2.8)$$

де A_0 – початкова амплітуда;

μ – коефіцієнт динамічної в'язкості середовища.

При цьому, для обчислення коефіцієнта динамічної в'язкості ідеального газу в залежності від температури може бути використана формула Сазерленда:

$$\mu(T) = \mu_0 \frac{T_0 + C}{T + C} \left(\frac{T}{T_0} \right)^{\frac{3}{2}}, \quad (2.9)$$

де T – задана температура в Кельвінах;

μ_0 – коефіцієнт в'язкості газу при температурі T_0 ;

T_0 – контрольна температура в Кельвінах;

C – стала Сазерленда для газу, в'язкість якого визначається.

Для повітря значення сталих становлять: $\mu_0 = 18,27 \cdot 10^{-6}$ Па·с,
 $T_0 = 291,15$ К, $C = 120$ К.

2.3. Розроблення цифрових фільтрів

Цифровий фільтр – математичний алгоритм, який вибірково змінює частотні складові вхідного цифрового сигналу. Наприклад, фільтр може затримувати одні частоти та пропускати інші.

Вибірковість фільтра за частотою можна розглядати як функцію залежності коефіцієнта передачі $K(f)$ – відношення амплітуд вихідного та вхідного сигналів – від частоти f . Така функція називається амплітудно-частотною характеристикою (АЧХ) фільтра. В залежності від вибірконості за частотою серед фільтрів можна виділити чотири основних класи:

- фільтр нижніх частот (ФНЧ, low-pass);
- фільтр верхніх частот (ФВЧ, high-pass);
- смуговий фільтр (СФ, band-pass);
- режекторний фільтр (РФ, band-stop).

На рис. 2.2 наведено графіки АЧХ так званих “ідеальних” фільтрів. В таких фільтрах наявна чітка границя між діапазоном частот, які затримуються та пропускаються фільтром. Такі діапазони називають смугами затримання (stopband) та пропускання (passband), відповідно.

Фільтр нижніх частот пропускає лише складові сигналу з частотою нижче заданої частоти зрізу f_c та придушує складові з вищими за неї частотами.

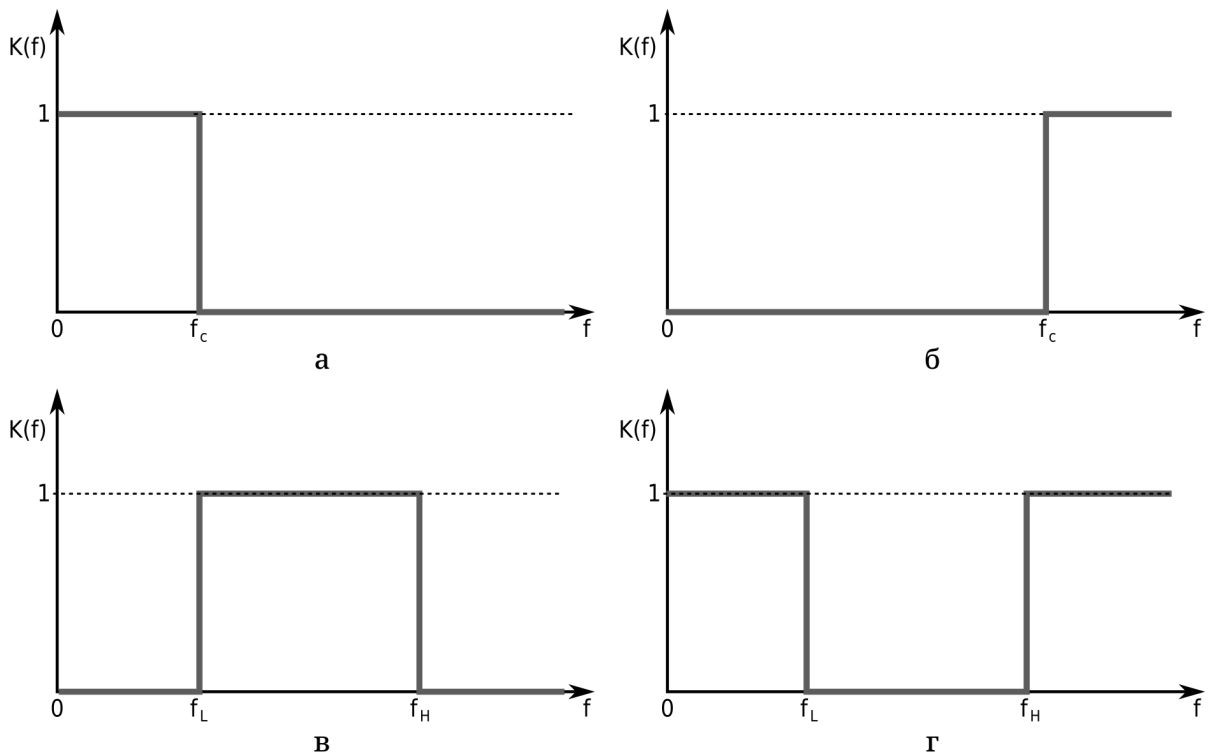


Рис. 2.2. АЧХ ідеальних фільтрів:

а) ФНЧ; б) ФВЧ; в) смугового фільтра; г) режекторного фільтра.

Фільтр верхніх частот, навпаки, послаблює низькочастотні складові сигналу та пропускає ті, частота яких вища за частоту зрізу f_c .

Смуговий фільтр пропускає складові сигналу, в яких частота знаходиться в межах між нижньою та верхньою граничною частотою смуги пропускання – f_L та f_H , відповідно. При цьому послаблює складові сигналу поза її межами.

Режекторний фільтр має дві смуги пропускання – нижню з верхньою граничною частотою f_L та верхню з нижньою граничною частотою f_H , між якими знаходиться смуга затримання.

Створення ідеального фільтра є фізично неможливим, тому використовують різноманітні апроксимації. В реальних фільтрах і у смугах затримання, і у смугах пропускання можуть бути наявні коливання, а між ними завжди наявна перехідні смуги. Ці відмінності графіків АЧХ реальних фільтрів продемонстровано на рис. 2.3.

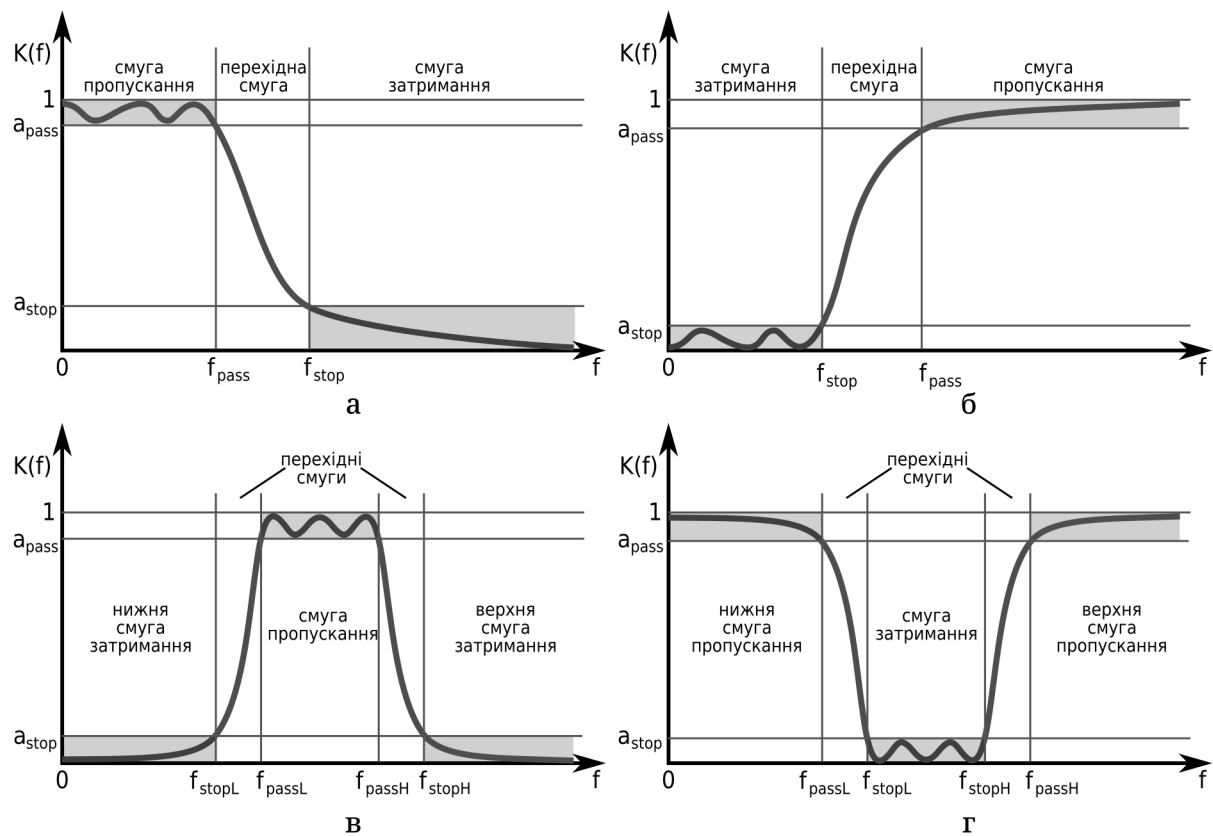


Рис. 2.3. АЧХ реальних фільтрів:

а) ФНЧ; б) ФВЧ; в) смугового фільтра; г) режекторного фільтра.

Вибір необхідної апроксимації завжди є компромісом – наприклад, мінімальне спотворення у смузі пропускання ціною більш широкої перехідної смуги.

Іншою важливою характеристикою цифрових фільтрів є залежність між фазою вхідного та вихідних сигналів – фазово-частотна характеристика (ФЧХ), яка є функцією частоти. Її лінійність означатиме однаковість часової затримки для всіх частотних складових сигналу і забезпечує відсутність фазових спотворень у вихідному сигналі [15, 16]. Слід зауважити, що такі спотворення для аудіосигналу можуть бути помітними при його прослуховуванні [17].

Розрізняють дві категорії цифрових фільтрів відповідно до їх імпульсної характеристики (реакції на одиничний імпульс):

- СІХ (FIR) – фільтри зі скінченною імпульсною характеристикою;
- НІХ (IIR) – фільтри з нескінченною імпульсною характеристикою.

Застосування фільтра до сигналу в часовій області представлено операцією згортки імпульсного відгуку фільтра з вхідним сигналом:

$$y[n] = \sum_{k=0}^N x[n-k]h[k], \quad (2.10)$$

де N – довжина послідовності імпульсного відгуку $h[n]$.

Обчислення згортки безпосередньо за формулою (2.10) є неефективним, оскільки для обчислення N вихідних значень необхідно виконати N^2 операцій, тому в практичних застосуваннях використовують алгоритми швидкої згортки, які знижують складність обчислень з $O(N^2)$ до $O(N \log N)$. Найбільш розповсюджені з них побудовані на алгоритмі швидкого перетворення Фур'є: до послідовностей вхідних значень та імпульсного відгуку застосовують перетворення, результати перетворення почленно перемножують і до отриманих значень застосовують зворотне перетворення Фур'є.

Для реалізації цифрових фільтрів достатньо трьох основних операцій: додавання, множення на константу та затримки. На рис. 2.4 наведено загальні структурні схеми для прямої форми реалізації СІХ- та НІХ-фільтрів.

Фільтр називають стійким, якщо його імпульсна характеристика має скінченну довжину або прямує до нуля з часом. Інакше поняття стійкості можна сформулювати наступним чином: фільтр є стійким тоді та тільки тоді, коли для будь-якого обмеженого в часі вхідного сигналу фільтр породжує обмежений в часі вихідний сигнал [18].

Крім того, фільтри, значення вихідного сигналу яких залежать лише від поточних та минулих значень вхідного сигналу називають каузальними, причинними або фізично реалізовними для систем, що оброблюють сигнал в режимі реального часу. Фільтри, вихідний сигнал яких залежить від майбутніх значень вхідного сигналу, можуть бути перетвореними на

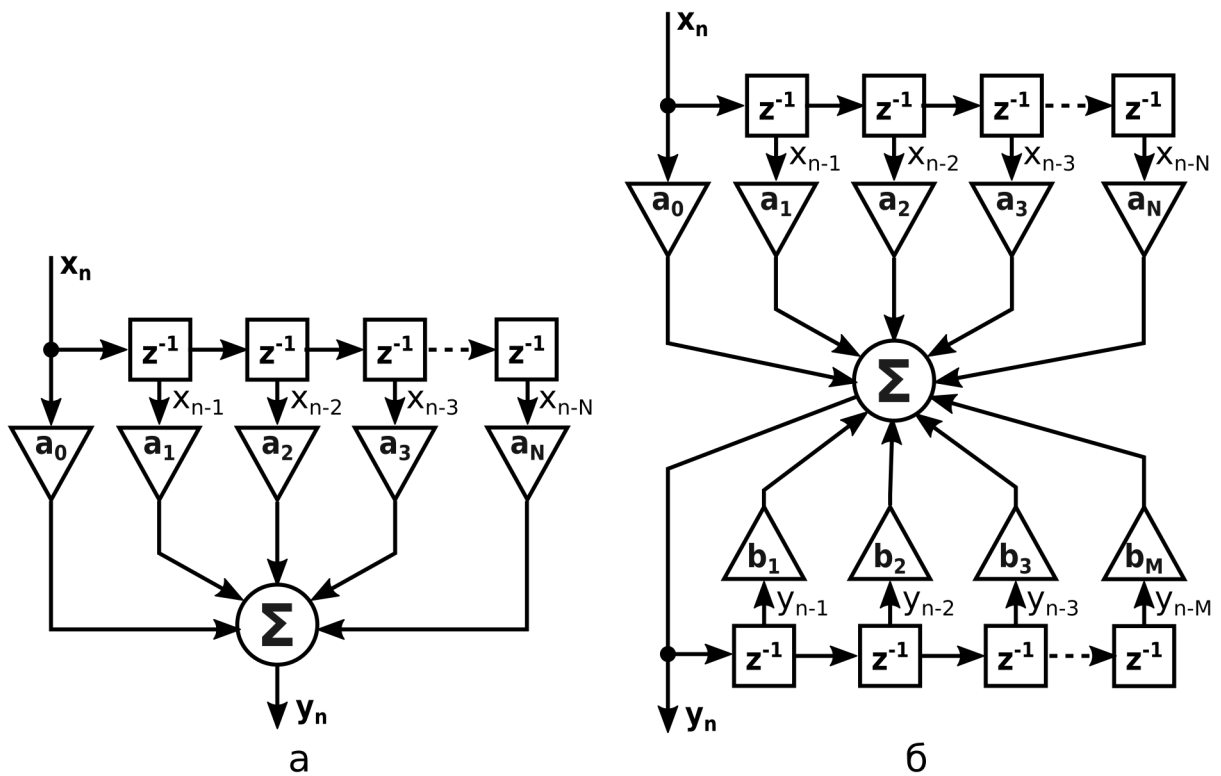


Рис. 2.4. Загальна структура прямої форми реалізації цифрових фільтрів:
а) нерекурсивного; б) рекурсивного. Трикутниками позначено коефіцієнти підсилення, квадратами – одиничну затримку вхідного сигналу.

каузальні за допомогою зсуву та/або відкидання некаузальної частини імпульсного відгуку.

СІХ-фільтри також називаються нерекурсивними, оскільки значення вихідного сигналу залежить лише від значень вхідного сигналу. Загальний вигляд різницевого рівняння подано у формулі (2.11), передавальної функції – у формулі (2.12). Число N визначає порядок фільтра. Імпульсна характеристика таких фільтрів є обмеженою в часі – з певного моменту часу відгук стає рівним нулю.

$$y[n] = a_0 x[n] + a_1 x[n-1] + \dots + a_N x[n-N] = \sum_{k=0}^N a_k x[n-k], \quad (2.11)$$

$$H(z) = \sum_{k=0}^N a_k z^{-k}. \quad (2.12)$$

СІХ-фільтри є завжди стійкими [19]. В більшості випадків вважають, що їх передавальна функція має лише нулі, оскільки всі полюси розміщені

у початку координат, $z=0$, і не мають впливу на стійкість фільтра. При цьому число полюсів дорівнює числу нулів. СІХ-фільтр матиме лінійну фазо-частотну характеристику за умови симетричності (або антисиметричності) коефіцієнтів a_k відносно центрального коефіцієнта, тобто $a_k = a_{N-k}$ або $a_k = -a_{N-k}$. При цьому найбільш універсальними є фільтри з непарним числом симетричних коефіцієнтів (парного порядку) – вони однаково підходять для реалізації ФНЧ, ФВЧ, смугових та режекторних фільтрів [20].

Імпульсна перехідна функція НІХ-фільтрів є теоретично необмеженою у часі. Значення вихідного сигналу такого фільтра залежать не лише від значень вхідного сигналу, а й від попередніх значень вихідного сигналу, утворюючи зворотний зв'язок. Через це НІХ-фільтри також називають рекурсивними. Загальний вигляд різницевого рівняння наведено у формулі (2.13), передавальної функції – у формулі (2.14). Порядок фільтра визначає максимальне з чисел N та M – порядок вхідного сигналу та порядок зворотного зв'язку, відповідно.

$$y[n] = a_0 x[n] + a_1 x[n-1] + \dots + a_N x[n-N] - (b_1 y[n-1] + b_2 y[n-2] + \dots + b_M y[n-M]) = \sum_{k=0}^N a_k x[n-k] - \sum_{k=1}^M b_k y[n-k], \quad (2.13)$$

$$H(z) = \frac{\sum_{k=0}^N a_k z^{-k}}{1 + \sum_{k=1}^M b_k z^{-k}}. \quad (2.14)$$

ФЧХ НІХ-фільтрів є нелінійною.

Для забезпечення стійкості НІХ-фільтрів необхідно, щоб всі полюси передавальної функції лежали всередині одиничного кола [18]. Якщо полюс лежить за його межами, фільтр буде нестійким. Полюси, що лежать на одиничному колі, дають нестійкі або потенційно нестійкі фільтри за виключенням випадків, коли полюси на одиничному колі співпадають з нулем, що призводить до компенсації їх впливу на стійкість.

З наближенням АЧХ до ідеального фільтра порядок НІХ-фільтра буде нижчим, ніж в аналогічного СІХ-фільтра, а це впливає на час обробки та витрати пам'яті. З метою зниження порядку СІХ-фільтрів застосовують алгоритми оптимізації.

В табл. 2.1 наведено порівняння СІХ- та НІХ-фільтрів, на основі його аналізу для оброблення аудіосигналу було обрано СІХ-фільтри.

Таблиця 2.1

Порівняння типів цифрових фільтрів

Тип фільтрів	Можливо гарантувати лінійність ФЧХ	Мінімальний порядок	Гарантована стійкість
СІХ	+	–	+
НІХ	–	+	–

Методи проєктування СІХ-фільтрів ґрунтуються на безпосередній апроксимації бажаної (ідеальної) АЧХ. Найпростіший з них – метод зважування (вагових коефіцієнтів, зворотного перетворення Фур'є), на ньому також базується віконний метод.

АЧХ ідеального фільтра нижніх частот має наступний вигляд:

$$H_D(e^{j\omega}) = \begin{cases} 1, & |\omega| \leq \omega_c, \\ 0, & |\omega| > \omega_c, \end{cases} \quad (2.15)$$

де ω_c – частота зрізу.

При цьому періодизовану (з періодом 2π) ідеальну АЧХ можна подати у частотній області наступним рядом Фур'є:

$$H_D(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h_D[n] e^{-j\omega n}, \quad (2.16)$$

де $h_D[n]$ – коефіцієнти ряду, які відповідають послідовності імпульсного відгуку та коефіцієнтам фільтра.

Імпульсний відгук також можна виразити через $H_D(e^{j\omega})$, застосувавши до неї зворотне перетворення Фур'є:

$$h_D[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_D(e^{j\omega}) e^{j\omega n} d\omega. \quad (2.17)$$

Після підстановки (2.15) у вираз (2.17) отримаємо:

$$h_D[n] = \frac{1}{2\pi} \int_{-\omega_c}^{\omega_c} e^{j\omega n} d\omega = \frac{\sin(n\omega_c)}{n\pi}. \quad (2.18)$$

Отримана імпульсна характеристика, приклад якої наведено на рис. 2.5, не є ні скінченною (хоча і поступово зменшується із віддаленням від точки $n=0$), ні каузальною. Найпростіший спосіб отримати скінченну імпульсну характеристику – використовувати лише N її значень, а всі інші вважати рівними нулю.

Враховуючи вищенаведені умови лінійності ФЧХ, N має бути парним, а обрані коефіцієнти – симетричними. При цьому каузальність фільтра може бути досягнута внесенням затримки тривалістю $\frac{N}{2}$ відліків до імпульсного відгуку ідеального фільтра. В результаті цих перетворень імпульсна характеристика реалізованого фільтра матиме вигляд:

$$h[n] = \begin{cases} h_D[n - \frac{N}{2}], & 0 \leq n \leq N \\ 0 & \text{інакше} \end{cases}. \quad (2.19)$$

Порядок фільтра N стає результатом компромісу між близькістю АЧХ реалізованого фільтра до ідеального та, наприклад, часовою затримкою, яку вносить фільтр, вартістю реалізації тощо.

АЧХ реалізованого фільтра зі збільшенням числа коефіцієнтів наближається до АЧХ ідеального, що проілюстровано на рис. 2.6. З порівняння отриманих графіків, видно також і те, що апроксимації мають помітні сплески у смугах затримки та пропускання. Вони отримали назву явища Гіббса, є наслідком апроксимації розривної функції рядом та не зникають зі збільшенням числа коефіцієнтів, а лише стають вужчими та концентруються біля перехідної смуги.

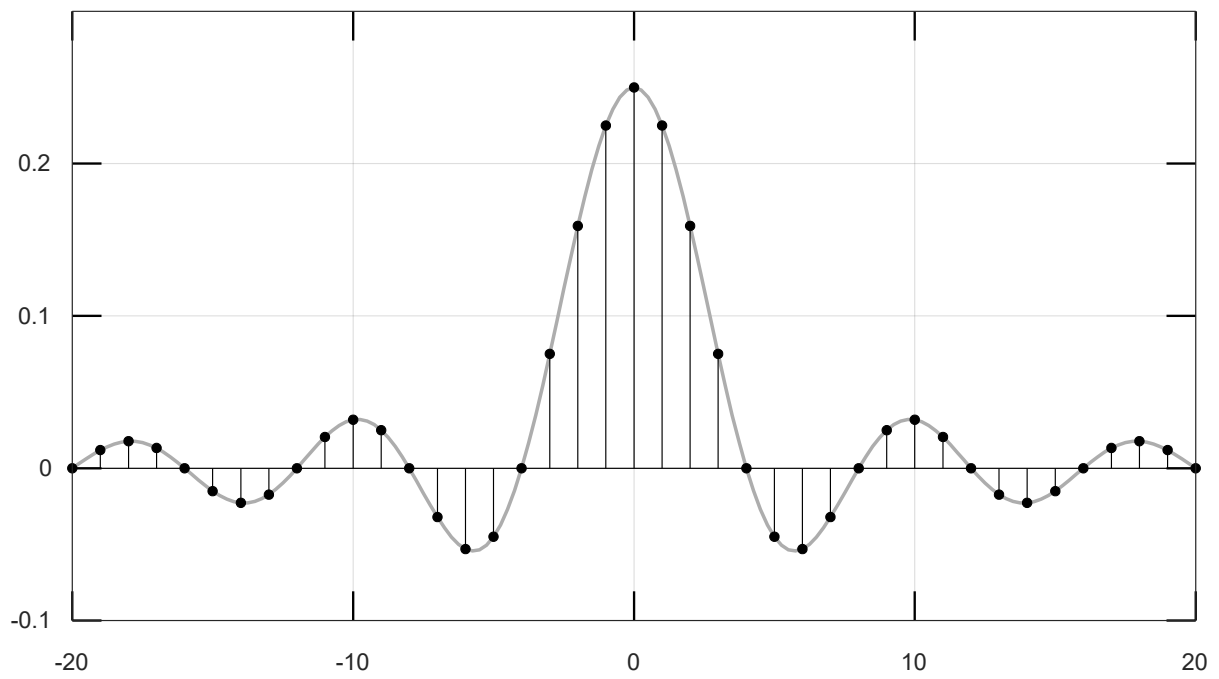


Рис. 2.5. Імпульсний відгук ідеального $\Phi НЧ$ з $\omega_c = \frac{\pi}{4}$

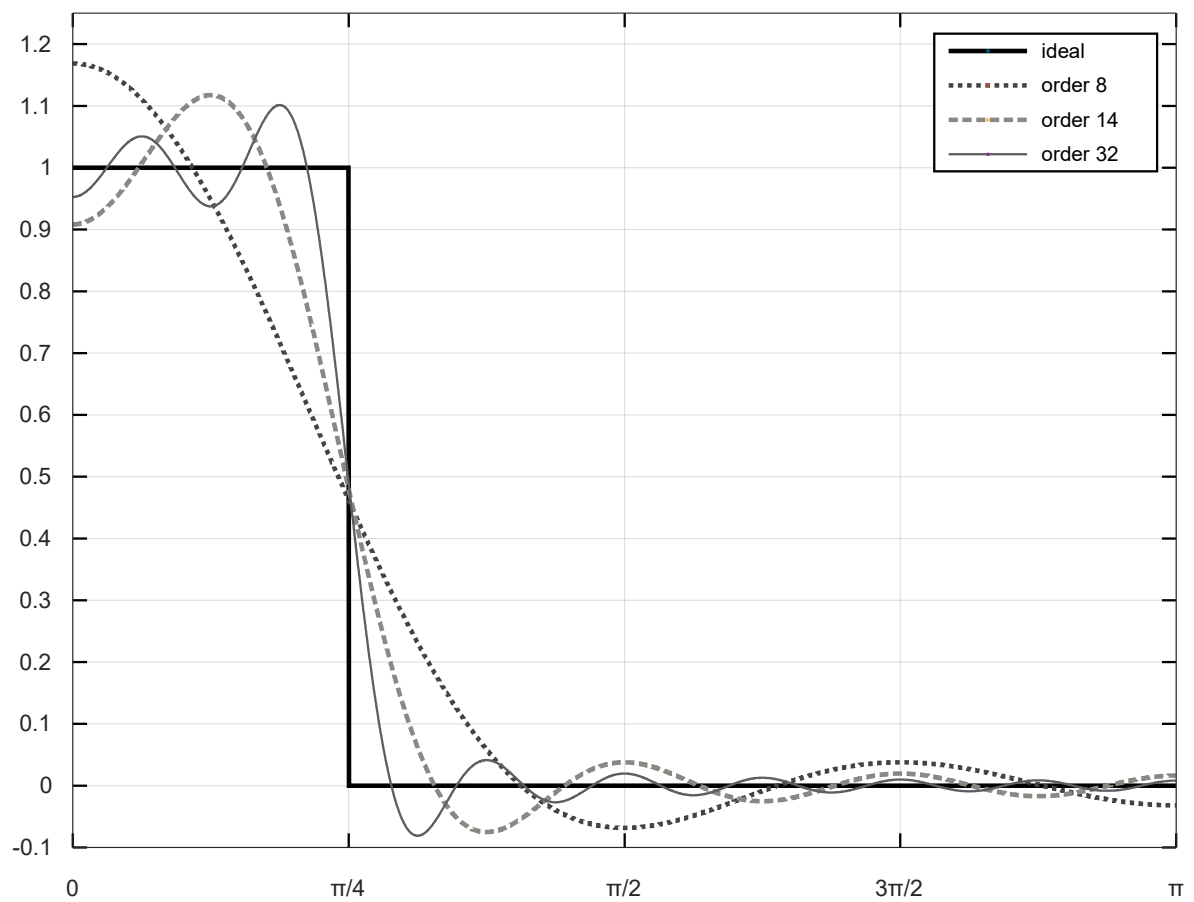


Рис. 2.6. Графік АЧХ ідеального $\Phi НЧ$ з $\omega_c = \frac{\pi}{4}$ та його апроксимацій

8, 14 та 32 порядків

Відкидання значень імпульсної характеристики $h_D[n]$ рівнозначне множенню ідеальної імпульсної характеристики на прямокутну вагову, або віконну, функцію:

$$w_{rect}[n] = \begin{cases} 1, & 0 \leq n \leq N, \\ 0. & \end{cases} \quad (2.20)$$

Для зменшення сплесків у смугах затримки та пропускання використовують альтернативні віконні функції, значення яких плавно спадають до нуля з обох боків “вікна”. Отримана ширша перехідна смуга може бути скомпенсована підвищенням порядку фільтра. До найбільш поширених вікон належать: трикутне (Бартлета) – (2.21), Хана – (2.22), Хеммінга – (2.23), Блекмена – (2.24). Їх графіки наведено на рис. 2.7.

$$w_{Bart}[n] = \begin{cases} 1 - \left| \frac{2n}{N} - 1 \right|, & 0 \leq n \leq N, \\ 0; & \end{cases} \quad (2.21)$$

$$w_{Hann}[n] = \begin{cases} 0,5 - 0,5 \cos \frac{2\pi n}{N}, & 0 \leq n \leq N, \\ 0; & \end{cases} \quad (2.22)$$

$$w_{Hamming}[n] = \begin{cases} 0,54 - 0,46 \cos \frac{2\pi n}{N}, & 0 \leq n \leq N, \\ 0; & \end{cases} \quad (2.23)$$

$$w_{Blackman}[n] = \begin{cases} 0,42 - 0,5 \cos \frac{2\pi n}{N} + 0,08 \cos \frac{4\pi n}{N}, & 0 \leq n \leq N, \\ 0. & \end{cases} \quad (2.24)$$

Особливої уваги заслуговує вікно Кайзера, яке є близьким до оптимального [18] та представлене формулою (2.25).

$$w_{Kaiser}[n] = \begin{cases} \frac{I_0 \left(\beta \sqrt{1 - \frac{(2n - N)^2}{N^2}} \right)}{I_0(\beta)}, & 0 \leq n \leq N, \\ 0, & \end{cases} \quad (2.25)$$

де $I_0(x)$ – модифікована функція Бесселя першого роду нульового порядку.

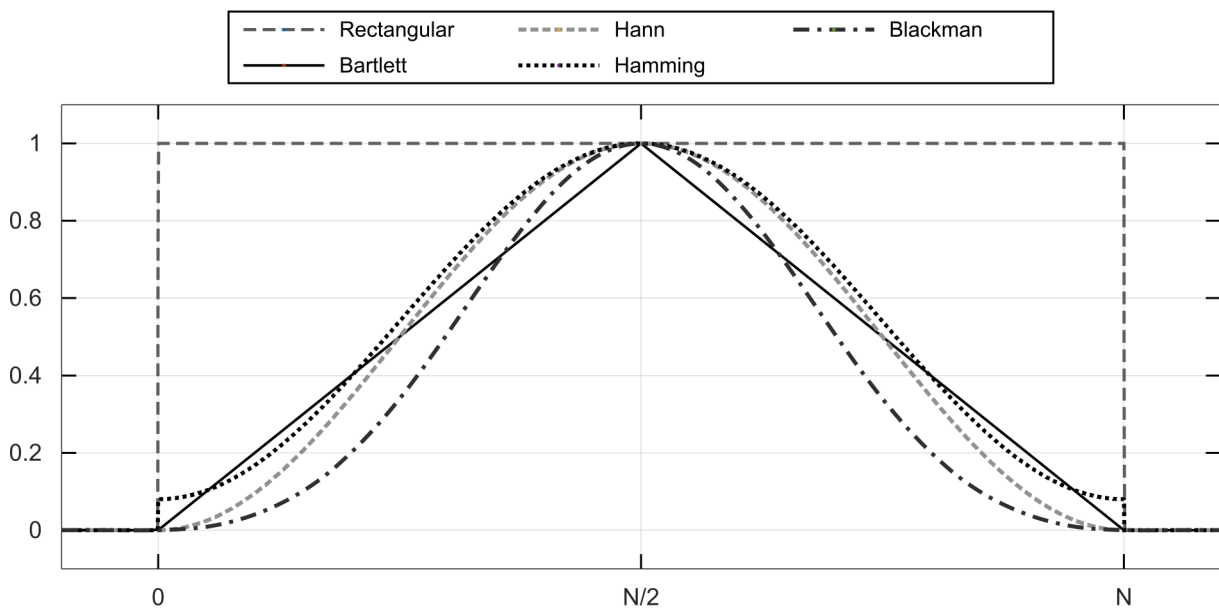


Рис. 2.7. Графіки віконних функцій: прямокутного, Бартлєта, Хана, Хеммінга та Блекмена

Значення параметра β залежить від максимальних відхилень δ_s та δ_p в смугах пропускання та затримки:

$$\beta = \begin{cases} 0,1102(A - 8,7), & A > 50, \\ 0,5842(A - 21)^{0,4} + 0,07886(A - 21), & 21 \leq A \leq 50, \\ 0 & A < 21, \end{cases} \quad (2.26)$$

де $A = -20 \log_{10}(\min[\delta_p, \delta_s])$.

Крім того, можливо отримати значення мінімально можливого порядку фільтра для заданих параметрів відхилень та ширини перехідної смуги:

$$N = \frac{A - 8}{2,285 \Delta \omega}, \quad (2.27)$$

де $\Delta \omega$ – ширина перехідної смуги $\Delta \omega$.

На основі коефіцієнтів $a_k^{(LP)}$ ФНЧ можливо отримати коефіцієнти $a_k^{(HP)}$ ФВЧ (2.28):

$$a_k^{(HP)} = \delta[k] - a_k^{(LP)} \quad (2.28)$$

де $\delta[n]$ – функція одиничного імпульсу.

Аналогічно можна отримати коефіцієнти смугового фільтра (2.29) та режекторного (2.30) на основі коефіцієнтів ФНЧ, частоти зрізу яких відповідатимуть верхній та нижній граничним частотам смуги пропускання (затримки):

$$a_k^{(BP)} = a_k^{(LP_H)} - a_k^{(LP_L)} \quad (2.29)$$

$$a_k^{(BS)} = \delta[k] - a_k^{(LP_H)} + a_k^{(LP_L)} \quad (2.30)$$

2.4. Формулювання способу ауралізації

Модельоване оточення подається як множина кімнат, які мають спільні конструктивні елементи-перегородки – стіни, вікна, двері, отвори тощо. Джерела звуку та слухач розташовуються в межах однієї з даних кімнат. На рис. 2.9 наведено приклад такого оточення. На одному поверсі розміщено чотири кімнати $R_1 - R_4$ зі спільними перегородками $p_1 - p_9$. Два джерела звуку $S_1 - S_2$ та слухач L розміщені у кімнатах R_1 , R_4 та R_3 , відповідно.

Елементи-перегородки визначаються матеріалом, розмірами та положенням. В наведеному прикладі перегородки p_2 та p_8 відповідають отворам в стінах (матеріал вважається повітрям), p_5 – дверям, всі інші – стінам.

Зв'язки кімнат оточення можна подати як неорієнтований мультиграф без петель, в якому ребрам відповідають елементи-перегородки. Граф зв'язків кімнат з прикладу зображено на рис. 2.8.

Джерела звуку вважаються точковими та неспрямованими, в сцені вони визначаються координатами положення та джерелом вихідного аудіосигналу. За необхідності в моделюванні напрямленості, визначення

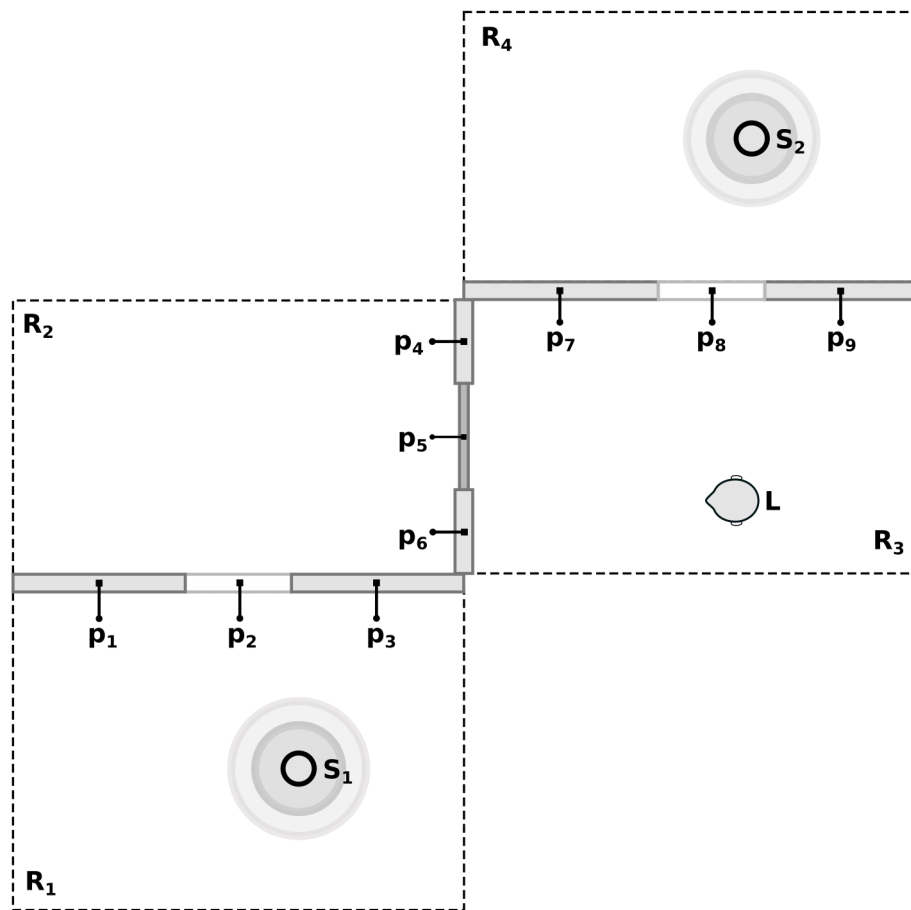


Рис. 2.9. Приклад модельованого оточення: чотири кімнати на одному поверсі з двома джерелами та слухачем

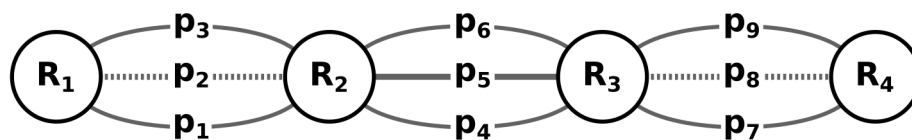


Рис. 2.8. Граф зв'язків кімнат з прикладу на рис. 2.9

джерела доповнюється функцією, яка визначає силу вихідного сигналу за азимутальним та полярним кутом.

Слухач представлений двома параметрами: координати положення та вектор орієнтації в просторі.

Для кожної пари джерело-слухач в графі відповідно до їх розміщення в кімнатах відшукуються всі можливі елементарні шляхи поширення звуку від обраного джерела до слухача. Для їх пошуку можливо застосувати алгоритми пошуку в глибину або пошуку в ширину. Наприклад, між

джерелом S_1 та слухачем L отримаємо наступні шляхи поширення:

$$R_1 \xrightarrow{p_1} R_2 \xrightarrow{p_4} R_3, \quad R_1 \xrightarrow{p_1} R_2 \xrightarrow{p_5} R_3, \quad R_1 \xrightarrow{p_1} R_2 \xrightarrow{p_6} R_3, \quad R_1 \xrightarrow{p_2} R_2 \xrightarrow{p_4} R_3, \quad R_1 \xrightarrow{p_2} R_2 \xrightarrow{p_5} R_3, \\ R_1 \xrightarrow{p_2} R_2 \xrightarrow{p_6} R_3, \quad R_1 \xrightarrow{p_3} R_2 \xrightarrow{p_4} R_3, \quad R_1 \xrightarrow{p_3} R_2 \xrightarrow{p_5} R_3, \quad R_1 \xrightarrow{p_3} R_2 \xrightarrow{p_6} R_3.$$

Кожен знайдений шлях використовується для обчислення графа передатних функцій (ГПФ). Наприклад, перший шлях дасть наступний оргграф: $H_{S_1 R_1 p_1} \rightarrow H_{p_1} \rightarrow H_{p_1 R_2 p_4} \rightarrow H_{p_4} \rightarrow H_{p_4 R_3 L}$, в якому $H_{X R_i Y}$ – передатна функція ділянки шляху між джерелом X та отримувачем Y в межах кімнати R_i , а H_{p_i} – передатна функція елемента-перегородки p_i . Перегородки, крізь які надходить звук, в суміжній кімнаті вважаються вторинними точковими джерелами, які розміщені в центрі площини перегородки; практичність такого підходу обґрунтована в [21].

Відношення положення джерела, яке знаходиться в кінцевій кімнаті шляху, до положення слухача може бути використано у відповідній передатній функції для відтворення просторового звуку, наприклад, з використанням бінауральної стереофонії. Таким чином, вихідний аудіосигнал може мати два і більше канали, але всі проміжні обчислення виконуються над монофонічним сигналом.

Загальна передатна функція шляху поширення представлена добутком всіх передатних функцій, що входять до ГПФ. Отримані показники затухання сигналу джерела використовуються для параметризації набору смугових (октавних) фільтрів.

Узагальнення основних кроків роботи способу наведено на діаграмі діяльності на рис. 2.10.

У програмах, які виконують оброблення сигналу у реальному часі та надають можливість змінювати положення слухачів та джерел, даний спосіб дозволяє застосовувати оптимізації при обчисленні сумарного відгуку. Наприклад, переміщення слухача в межах однієї кімнати не вимагає повторного пошуку шляхів поширення та побудови ГПФ заново,

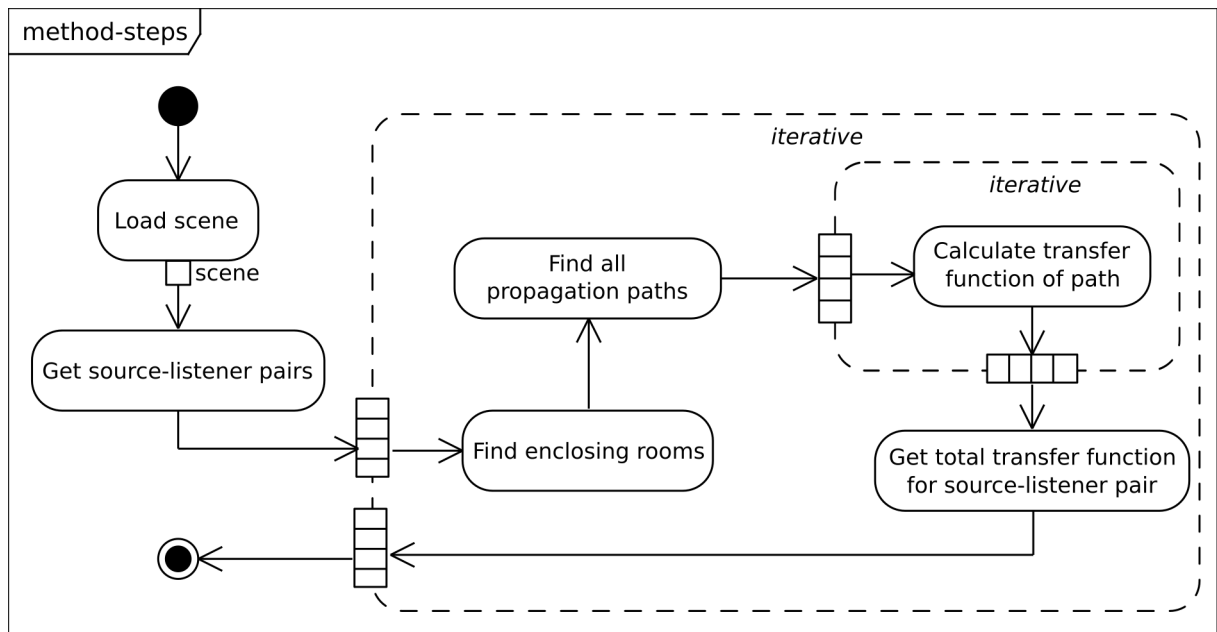


Рис. 2.10. Основні кроки роботи способу

достатньо лише переобчислити значення останньої передатної функції в ньому. Аналогічно, при переміщенні джерела в межах однієї кімнати достатньо переобчислити значення першої функції.

Розроблений спосіб враховує лише прямі шляхи поширення шуму без виникнення дифракції та реверберації, а перегородки-випромінювачі звуку вважаються вторинними точковими джерелами. Такі спрощення зроблено з метою зменшити кількість необхідних обчислень та дозволити оброблення аудіосигналу в реальному часі з якомога мінімальною втратою реалістичності звучання.

2.5. Висновки до розділу 2

В даному розділі сформульовано науково-інноваційну задачу, проведено аналіз математичних моделей поширення звуку в будівлях та розглянуто підходи до створення цифрових фільтрів для оброблення аудіосигналу.

На основі теоретичних відомостей, наведених в попередньому та поточному розділах, запропоновано спосіб ауралізації поширення повітряного шуму в межах множини кімнат будівлі від джерел до слухача.

Модельоване оточення подається як мультиграф без петель, в якому вузлами є кімнати, а ребрами – конструктивні елементи-перегородки. Пошук шляхів поширення звуку та їх передатні функції дозволяє обчислити імпульсний відгук середовища, який буде застосовано до вихідних сигналів джерел звуку множиною октавних фільтрів. Спосіб може бути розширеним для отримання просторового звучання та надає можливості для оптимізації обчислень.

3. РОЗРОБЛЕННЯ ІННОВАЦІЙНОГО ПРОГРАМНОГО ПРОДУКТУ

Розвиток інноваційних продуктів починається зі створення мінімально життєздатного продукту (MVP), який дозволить отримати зворотний зв'язок від користувачів необхідний для подальшого розвитку. Такий продукт надає обмежену функціональність, яка при цьому має максимально відповідати потребам ранніх користувачів у вирішенні їх ключових проблем.

Отже, перед нами стоїть задача створення MVP на основі способу розробленого в підрозділі 2.4.

3.1. Вимоги до програмного продукту

Аналіз зацікавлених сторін, наведених в підрозділі 5.2, дозволяє нам сформулювати вимоги до створюваного програмного забезпечення.

3.1.1. Функціональні вимоги

Система повинна надавати користувачу наступні можливості:

- перемикатися між двома режимами роботи:
 - режим редагування параметрів оточення;
 - режим симуляції;
- в режимі редагування параметрів оточення:
 - створювати нові, виконувати збереження відкритих та завантажувати наявні файли сцен;
 - створювати та/або імпортувати тривимірні моделі приміщень;
 - переглядати тривимірну модель сцени із розміщеними в ній кімнатами, елементами-перегородками, слухачем та джерелами;
 - переглядати перелік джерел звуку, кімнат та елементів-перегородок наявних в сцені, видаляти їх та додавати нові;
 - переглядати базу даних типових та користувацьких визначень матеріалів;

- додавати, редагувати та видаляти користувацькі визначення матеріалів;
- призначати спільні перегородки для суміжних кімнат;
- редагувати розміщення та параметри кімнат і перегородок;
- редагувати розміщення та параметри джерел і слухача;
- в режимі симуляції – прослухати результати моделювання із переглядом.

Система повинна сповіщати користувача про некоректність заданих налаштувань.

Діаграма варіантів використання в нотації UML, подана на рис. 3.1, є узагальненням наведених вище функціональних вимог.

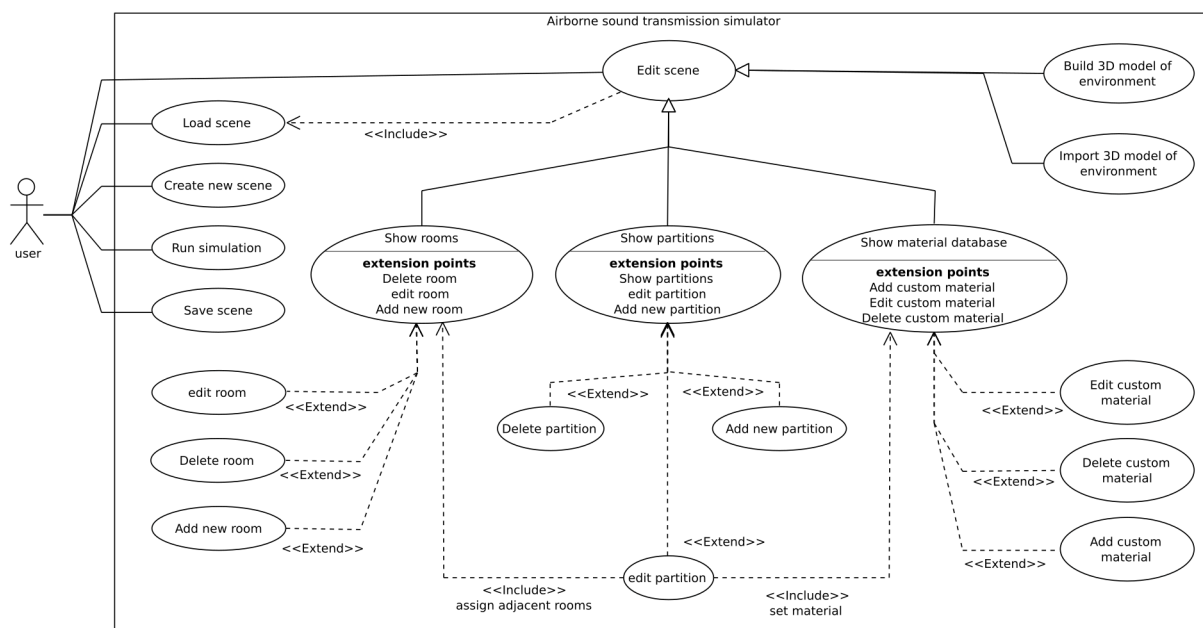


Рис. 3.1. Варіанти використання програмного продукту

3.1.2. Нефункціональні вимоги

Продукт повинен забезпечувати:

- графічний інтерфейс для взаємодії з користувача з програмним забезпеченням;
- інтеграцію з наявними редакторами тривимірних моделей;

- підтримку основних ОС для персональних комп'ютерів, першочергово – Microsoft Windows 8.1 та Windows 10.

Компоненти, які відповідають безпосередньо за обчислення імпульсного відгуку середовища та оброблення аудіосигналу повинні бути виділені у модуль, який компілюється в машинний код.

3.2. Вибір засобів розроблення

До редакторів тривимірних моделей, які є популярними в архітектурній візуалізації та надають підтримку користувацьких розширень належать: Blender, серія продуктів компанії Autodesk (3ds Max, Revit, AutoCAD) та SketchUp. Кожен з них надає SDK для розробників розширень та відповідну документацію.

Продукти компанії Autodesk мають високу вартість ліцензій – від 1545 до 2310 доларів на рік [22]. Це змушує відмовитися від створення розширення для будь-якого з них в першу чергу, не зважаючи на їх популярність. Іншим недоліком є закритість екосистеми розробників розширень. Всі з названих продуктів доступні для Microsoft Windows, AutoCAD додатково має версію для MacOS X.

SketchUP також є платним, проте зі значно нижчою вартістю підписки на рік (базова версія з підтримкою розширень – 299 доларів [23]), що робить даний продукт кандидатом на розроблення розширення для нього в майбутньому. SketchUp надає підтримку Microsoft Windows та macOS X.

Blender є вільним та безкоштовним програмним забезпеченням із відкритою екосистемою розробників. Програма надає всіх основних операційних систем для персональних комп'ютерів. Популярність Blender помітно зростає в останні роки, відбувається активний розвиток проєкту, значно вдосконалено інтерфейс користувача та розширено функціональність.

Альтернативою до створення розширень для редакторів тривимірних моделей є створення самостійної програми на основі рушіїв для створення мультимедійних прикладних програм з підтримкою 3D графіки та систем доповненої реальності. Найбільш популярними з тих, що застосовуються і для архітектурної візуалізації, є UnrealEngine та Unity. Їх використання в комерційних проєктах можливе без виплати роялті за певних умов.

Обидва продукти мають активну спільноту розробників, надають власні програми-редактори для роботи над розроблюваними проєктами. Серед підтримуваних цільових платформ – Windows, Linux та MacOS.

Тривимірні моделі оточення, які використовуються для візуалізації, можуть бути попередньо створені в окремому тривимірному редакторі. UnrealEngine підтримує імпорт моделей у двох поширених форматах – FBX та OBJ, а також відносно новий формат glTF. Unity, окрім імпорту даних форматів, надає можливість імпортувати власні формати деяких редакторів тривимірних моделей (наприклад, Blender та Maya). Крім цього, редактори кожного з рушіїв надають власні засоби для редагування полігональних моделей оточення.

Істотними перевагами Unity перед UnrealEngine є повнота документації та простота інтеграції користувацьких розширень в підсистему аудіо.

Таблиця 3.1

Порівняння тривимірних редакторів

		Autodesk 3ds Max, Revit, AutoCAD	SketchUp	Blender
Початкова вартість створення продукту на їх основі		\$1545...2310	\$299	0
Відкритість екосистеми розробників розширень		–	+/-	+
Підтримка ОС	Windows	+	+	+

Продовження табл. 3.1

		Autodesk 3ds Max, Revit, AutoCAD	SketchUp	Blender
Підтримка ОС	MacOS X	+/-	+	+
	Linux	—	—	+
Повнота документації для розробників		+/-	+/-	+/-
Простота інтеграції в підсистему аудіо		—	—	—

Таблиця 3.2

Порівняння програмних рушіїв

		Unity	UnrealEngine
Початкова вартість створення продукту на їх основі		0	0
Відкритість екосистеми розробників		+	+/-
Підтримка ОС	Windows	+	+
	MacOS X	+	+
	Linux	+	+
Імпорт поширених форматів тривимірних моделей		+	+
Повнота документації для розробників		+	—
Простота інтеграції в підсистему аудіо		+	—

Для програмної реалізації способу було обрано програмний рушій Unity. Логіка розроблюваних програм в ньому типово реалізовується мовою C# із використанням компонентного підходу із можливістю створення розширень мовою C++.

Для забезпечення керування версіями обрано систему Git.

3.3. Архітектура програмного продукту

На рис. 3.2 подано структуру та зв'язки модулів програмного забезпечення, що реалізує розроблений спосіб. У його архітектурі можна виділити три основних модулі:

- Spatializer Plugin;
- модуль компонентів об'єктів сцени;
- розширення інтерфейсу редактора Unity.

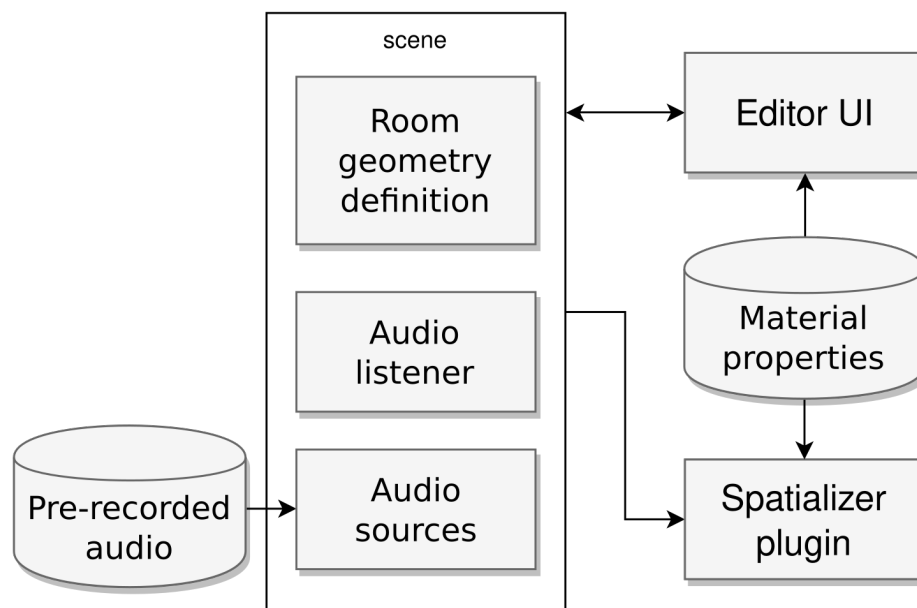


Рис. 3.2. Архітектура системи

Ядром програми, яке виконує перетворення вихідного аудіосигналу та обчислення передатних функцій поширення звуку від джерел звуку до слухача відповідно до способу, є модуль *Spatializer Plugin*. Його побудовано у вигляді динамічно завантажуваної бібліотеки – розширення Unity, яке реалізує інтерфейс Audio Spatializer SDK [25]. Даний SDK дозволяє стороннім розробникам інтегрувати власні реалізації звукових ефектів або засоби відтворення просторового звуку. У зв'язку з підвищеними вимогами до швидкодії таких модулів їх реалізують із використанням мови C++. З метою автоматизації процесу компіляції розширення в проєкті використовується система збирання CMake.

Підтримка бінауральної стереофонії забезпечено використанням HRTF на основі вільного набору даних вимірів KEMAR [24].

Модуль компонентів об'єктів сцени є проміжним шаром, необхідним для взаємодії інших двох модулів. Він надає дані, які доповнюють об'єкти сцени рушія Unity властивостями та зв'язками, специфічними для реалізації способу. До модуля входить набір компонентів, що безпосередньо призначаються об'єктам сцени, серед яких:

- PartitionBounds та RoomBounds – для визначення меж геометрії оточення, яка бере участь у симуляції, матеріалу перегородки;
- AudioListenerTarget та AudioSourceListener – розширення стандартних компонентів рушія, що представляють слухача та джерело звуку, необхідні для надання додаткової інформації про них ядру програми та користувачу.

Для зберігання та спільного доступу до визначень матеріалів використовуються файли ресурсів із серіалізованими об'єктами класу PartitionMaterialProperties, який є нащадком стандартного класу рушія ScriptableObject.

Набір попередньо записаного аудіо надає вихідний аудіосигнал для джерел звуку в сцені.

Модуль інтерфейсу редактора надає додаткові елементи графічного інтерфейсу в редакторі рушія Unity необхідні користувачеві для редагування параметрів компонентів *модуля сцени*: панелі інструментів та меню, маніпулятори тривимірних об'єктів, виведення допоміжної інформації.

3.4. Алгоритм пошуку елементарних шляхів в мультиграфах

Відповідно до описаного способу в підрозділі 2.4, для кожної пари джерело-слухач необхідно знайти всі наявні елементарні шляхи в графі зв'язків кімнат. Існує два базових алгоритми пошуку в скінченних графах: пошук в глибину (DFS) та пошук в ширину (BFS). Обчислювальна

складність обох алгоритмів $O(|V|+|E|)$, де $|V|$ – число вершин, $|E|$ – число ребер графа.

Виконання пошуку в глибину можна описати наступними кроками:

- 1) обирається початкова вершина;
- 2) позначити поточну вершину відвіданою;
- 3) якщо серед суміжних до поточної вершин є невідвідана, призначаємо її поточною та переходимо до кроку 2;
- 4) повернутися до вершини, з якої прийшли до поточної;
- 5) якщо повернулися до початкової вершини – робота алгоритму завершується, інакше – переходимо до кроку 3.

Пошук в ширину відвідує всі вершини на відстані k від початкової перш, ніж перейти до вершин на відстані $k+1$. Це можливо реалізувати із використанням черги обходу:

- 1) до черги додається обрана початкова вершина;
- 2) якщо черга порожня – робота алгоритму завершується, інакше – перша в черзі вершина стає поточною та видаляється із черги;
- 3) поточну вершину позначається відвіданою;
- 4) в кінець черги додаються всі невідвідані суміжні до поточної вершини та переходимо до кроку 2.

Різницю в порядку обходу графа для кожного з алгоритмів наведено на рис. 3.3.

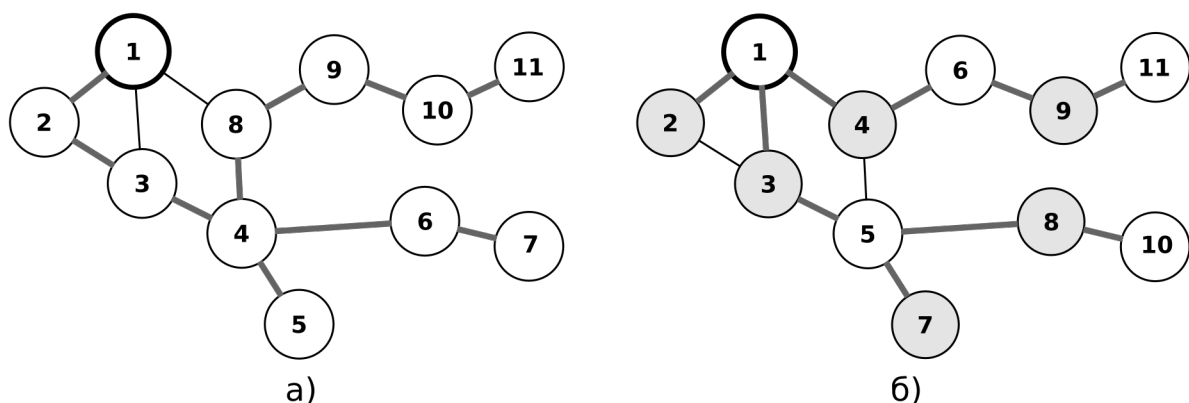


Рис. 3.3. Порядок обходу вершин графа:

а) пошук в глибину; б) пошук в ширину.

Діаграму діяльності розробленого алгоритму пошуку всіх шляхів в мультиграфі зв'язків кімнат подано на рис. 3.4, а його псевдокод – на лістингу 3.1.

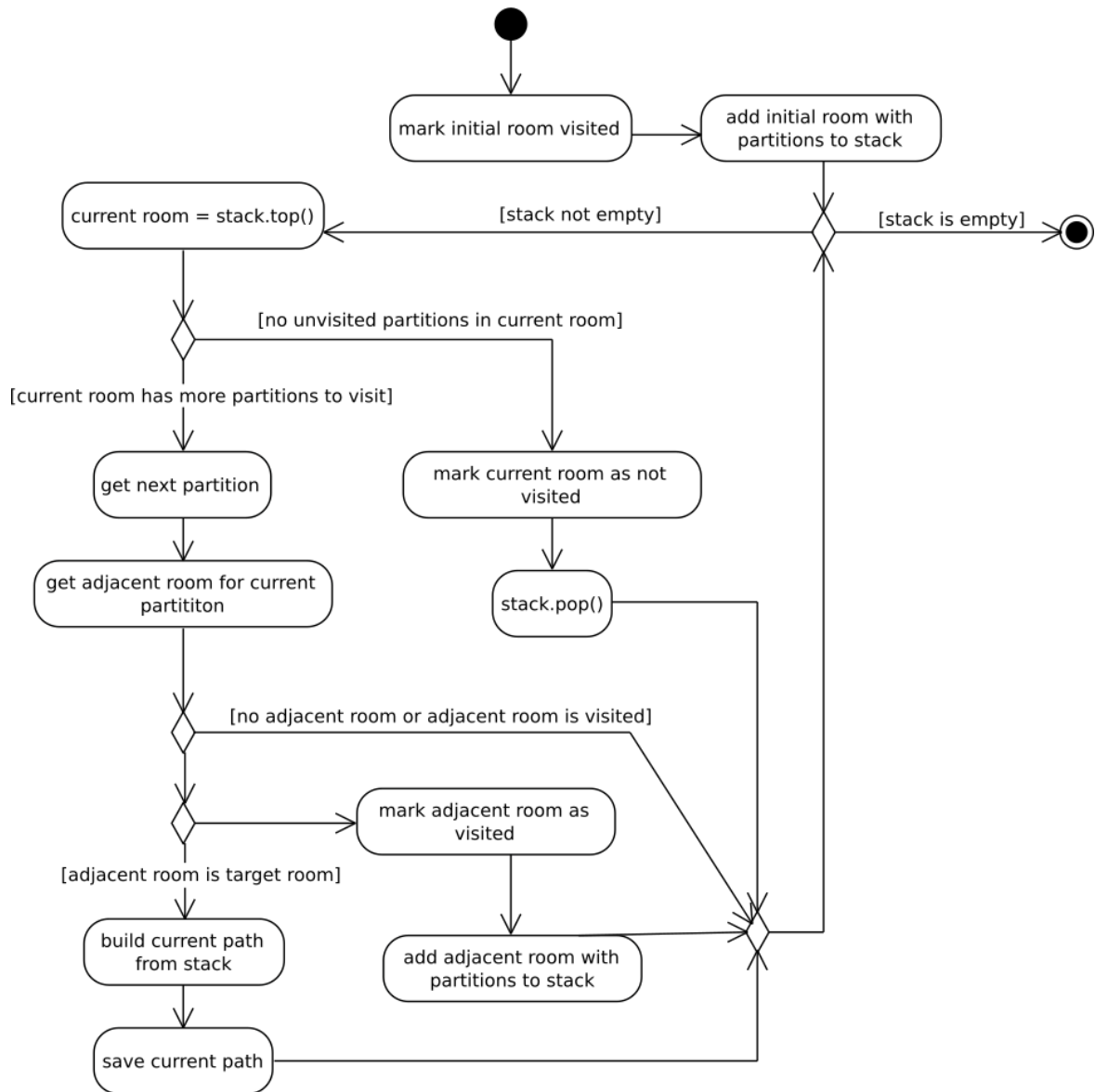


Рис. 3.4. Діаграма діяльності алгоритму пошуку всіх шляхів в мультиграфі

Лістинг 3.1. Псевдокод алгоритму пошуку всіх шляхів у мультиграфі

```

1 function get_all_paths_DFS(room_from, target_room):
2     room_from.is_visited = true
3     paths = []
4     room_stack = { room_from.get_partitions() }
5     while room_stack is not empty:
6         partition_queue = room_stack.peek()

```

Продовження лістингу 3.1

```
7         if partition_queue is empty:
8             current_room.is_visited = false
9             room_stack.pop()
10            continue
11        partition = partition_queue.get_next()
12        adjacent_room = partition.get_adjacent_room()
13        if adjacent_room is not set:
14            continue
15        if adjacent_room.is_visited:
16            continue
17        if adjacent_room is target_room:
18            build current_path from room_stack
19            paths.add(current_path)
20            continue
21        adjacent_room.is_visited = true
22        room_stack.push(room_from.get_partitions())
23    return paths
```

Для подання графа зв'язків кімнат використовуються списки суміжності. Алгоритм побудовано на основі пошуку в глибину. Використання пошуку в ширину в даному випадку вимагатиме використання додаткової пам'яті без очевидних переваг, оскільки необхідно відстежувати всі проміжні шляхи, в тому числі – ті, що не ведуть до цільової вершини.

3.5. Висновки до розділу 3

В даному розділі сформульовано основні вимоги до розроблюваного програмного продукту, який реалізує спосіб ауралізації, розроблений в підрозділі 2.4. Виявлення вимог проводилося із врахуванням необхідності створення мінімально життєздатного продукту, який орієнтовано на зацікавлених сторін в підрозділі 5.2.

Розглянуто два варіанти виконання програмного продукту: як розширення для редакторів тривимірних моделей або як окремої прикладної програми, побудованої з використанням програмних рушіїв з

підтримкою 3D графіки та систем доповненої реальності. На основі порівняльного аналізу засобів розроблення, основою для реалізації став рушій Unity.

Наведено розроблену архітектуру програмного продукту та ключові алгоритми. Проєкт реалізовано з використанням мов програмування C# для розширення редактора та C++ для ядра системи.

4. ТЕСТУВАННЯ ІННОВАЦІЙНОГО ПРОГРАМНОГО ПРОДУКТУ ТА АНАЛІЗ РОЗРОБЛЕНОГО СПОСОБУ

4.1. Методика тестування

Головна мета тестування – перевірка відповідності розробленого програмного продукту вимогам, висунутим до нього в підрозділі 3.1.

Тестування даного програмного продукту виконується методом напівавтоматизованого Gray Box Testing.

Для автоматизації процесів забезпечення якості ПЗ використовувалися наступні засоби:

- середовища модульного тестування NUnit та Catch2;
- утиліти статичного аналізу коду – StyleCop, Gendarme, CppCheck;
- Valgrind – інструмент профілювання використання пам'яті в C++.

Ручне тестування застосовується до інтерфейсу користувача та продукту загалом.

4.2. Створення тестових сценаріїв для ручного тестування

Приклад основних тестових сценаріїв, розроблених для ручного тестування програмного продукту наведено у табл. 4.1.

Таблиця 4.1

Тестові сценарії ручного тестування

№ з/п	Кроки тестового сценарію	Результат перевірки
1	Програма запускається без помилок	пройдено
2	При запуску програми початковим режимом є редагування сцени	пройдено
3	У головному вікні програми наявна кнопка-перемикач режиму роботи програми	пройдено
4	Надається візуальна індикація поточного режиму програми	пройдено

Продовження табл. 4.1

№ з/п	Кроки тестового сценарію	Результат перевірки
В режимі симуляції:		
5	Натискання кнопки-перемикача режиму програми переводить програму режим редагування сцени	пройдено
6	Вікно перегляду може бути максимізованим	пройдено
7	У вікні перегляду візуалізується тривимірне оточення з перспективи камери слухача на сцені	пройдено
8	Переміщення вказівника миші по вертикалі змінює полярний кут напрямку погляду віртуальної камери	пройдено
9	Переміщення вказівника миші по горизонталі змінює азимутальний кут напрямку погляду віртуальної камери	пройдено
10	Виконується відтворення налаштованих користувачем джерел звуку	пройдено
11	Зміна напрямку погляду камери відповідно впливає на отриманий стереофонічний звук	пройдено
12	Наявний бінауральний ефект, який дозволяє точно локалізувати джерела звуку навіть за відсутності візуальної інформації про їх розміщення	частково
13	Вибір розташування та матеріалів перегородки впливають на звук від джерел	в більшості випадків
В режимі редагування сцени:		
14	Натискання кнопки-перемикача режиму програми переводить програму режим симуляції	пройдено
15	Можливо створити нову сцену вибором відповідного пункту меню “Файл” або контекстного меню вікна файлової системи проєкту	пройдено
16	Можливо відкрити наявну сцену у вікні файлової системи проєкту	пройдено
17	Збереження змін виконується вибором відповідного пункту меню “Файл” або натисканням клавіатурного скорочення “Ctrl+S”	пройдено
18	За наявності незбережених змін редактор надає попередження користувачеві при закритті	пройдено

№ з/п	Кроки тестового сценарію	Результат перевірки
19	У вікні ієрархії сцени можливо відфільтрувати об'єкти за прикріпленими компонентами введенням назви компонента	пройдено
20	У вікні ієрархії сцени можливо відфільтрувати об'єкти за іменем	пройдено
21	Для створення нового об'єкта сцени слід обрати відповідний пункт з контекстного меню у вікні ієрархії або перегляду сцени	пройдено
22	Об'єктам сцени є можливість задати поворот, позицію та масштабування у вікні інспектора введенням дробових значень у відповідних полях	пройдено
23	Для зміни повороту, позиції та масштабування об'єктів у вікні перегляду сцени наявні відповідні маніпулятори	пройдено
24	Зміна типу маніпулятора об'єктів у вікні перегляду сцени відбувається натисканням відповідної кнопки-перемикача на панелі інструментів	пройдено
25	Обрані у вікні ієрархії або перегляду сцени об'єкти сцени видаляються натисканням клавіші "Delete" або вибором відповідного пункту контекстного меню	пройдено
26	Видалення об'єктів сцени автоматично видаляє і прикріплені до них компоненти	пройдено
27	Для прикріплення компонентів до обраних об'єктів наявна відповідна кнопка з меню у вікні інспектора	пройдено
28	При прикріпленні залежного компонента до об'єкту автоматично прикріплюється відсутній компонент-залежність: AudioSource – для AudioSourceTarget, AudioListener – для AudioListenerTarget	пройдено
29	Неможливо вилучити прикріплений компонент з об'єкта без видалення компонента, який від нього залежить	пройдено
30	Наявне вікно редагування сцени з попереднім переглядом	пройдено
31	Наявна можливість імпортувати в сцену тривимірну модель у форматах FBX та OBJ	пройдено

№ з/п	Кроки тестового сценарію	Результат перевірки
32	Наявні засоби створення полігональних моделей на сцені засобами редактора	пройдено
33	В інспекторі обраного джерела звуку можливо обрати джерело аудіосигналу з бази даних попередньо записаного аудіо	пройдено
34	В інспекторі обраного джерела звуку надається інформація про кімнату, в межах якої воно розміщене або найближчої до нього – в останньому випадку слід виводиться попередження “Джерело знаходиться поза межами кімнат”	пройдено
35	Положення джерел звуку та слухача відповідає положенню об’єктів, до яких прикріплено відповідні компоненти	пройдено
36	Положення перегородки або кімнати на сцені визначається положенням об’єкта сцени, до якого прикріплені відповідні компоненти, та відносним зсувом	пройдено
37	У вікні перегляду сцени межі перегородок та кімнат повинні бути приховані або показані в залежності від стану відповідного прапорця у вікні інспектора	пройдено
38	У вікні перегляду сцени межі перегородок та кімнат можуть бути змінені або перетягуванням кутів, або перетягуванням центральних точок сторін, для цього передбачено вибір режиму у вікні інспектора	пройдено
39	На сцені межі перегородок та кімнат показуються разом з їх іменем або іменем батьківського об’єкта, якщо не задано	пройдено
40	На сцені межі перегородок та кімнат показуються або з заданим для них кольором, або білим кольором, якщо прапорець “Використовувати колір границь” знято	пройдено
41	Вмикання маніпуляторів зміни меж перегородок та кімнат на сцені виконується встановленням відповідного прапорця у вікні інспектора	пройдено

№ з/п	Кроки тестового сценарію	Результат перевірки
42	Вмикання маніпуляторів зміни меж перегородок та кімнат на сцені приховує стандартні маніпулятори об'єктів у вікні сцени	пройдено
43	Зміна розмірів та відносного положення перегородки та кімнати може бути задане введенням дробових значень у відповідні поля у вікні інспектора	пройдено
44	У вікні інспектора для перегородки можливо обрати матеріал із бази даних властивостей матеріалів	пройдено
45	У вікні інспектора для перегородки можливо обрати з переліку наявних кімнат, до яких належить перегородка	пройдено
46	Для перегородки можливо задати не більше двох різних кімнат, до яких вона належить	пройдено
47	Скасувати належність перегородки до кімнат можна натисканням клавіші "Delete" у відповідному полі вікна інспектора	пройдено
48	Одинарне клацання лівої клавіші миші по полю заданої кімнати для перегородки у вікні інспектора підсвічує обрану кімнату у вікні ієрархії	пройдено
49	У вікні файлової системи проєкту наявний перелік файлів стандартних та користувацьких файлів опису властивостей матеріалів	пройдено
50	Для вибраного файлу опису властивостей матеріалів у вікні інспектора відображаються задані октавні коефіцієнти звукопроникності	пройдено
51	Редагування значень октавних коефіцієнтів звукопроникності у вікні інспектора зберігається у файлі обраного користувацького опису властивостей матеріалів	пройдено
52	Наявна можливість створення нового файлу користувацького опису властивостей матеріалу у вікні файлової системи проєкту	пройдено

4.3. Створення тестових моделей приміщень

Тривимірні моделі оточення можуть бути попередньо створені в будь-якому редакторі (наприклад, Blender), після чого їх достатньо імпортувати в редакторі рушія. Альтернативою такого підходу є використання інструмента ProBuilder, який дозволяє створювати полігональні моделі безпосередньо в редакторі Unity. Оскільки не було необхідності в створенні високоякісної візуалізації, використовувався вбудований інструмент.

Інтерфейс редактора показано на рис. 4.1, у вікні перегляду сцени показано модель оточення, яка відповідає прикладу з підрозділу 2.4. Компоненти розширення розташовані в підменю *Sound Transmission* стандартного меню *Components* та кнопки *Add Component* інспектора для обраного об'єкта.

Для створення тестової моделі приміщень необхідно виконати наступні кроки в редакторі Unity:

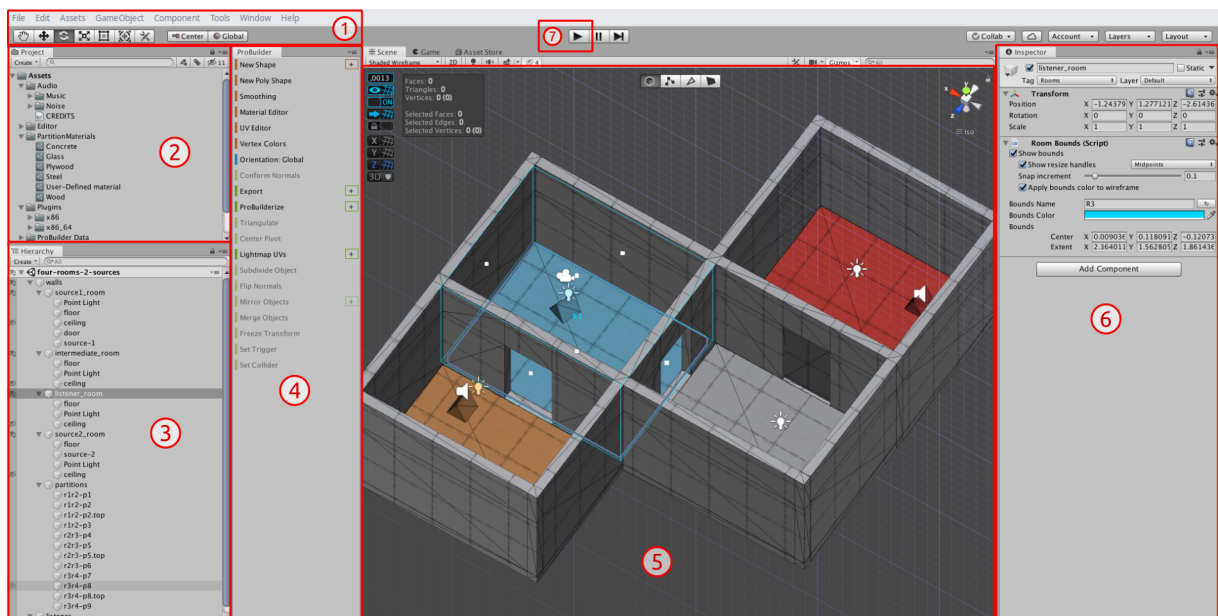


Рис. 4.1. Інтерфейс редактора Unity:

- 1 – панель інструментів та меню; 2 – панель файлової системи проєкту;
- 3 – панель ієрархії об'єктів сцени; 4 – панель інструментів ProBuilder;
- 5 – вікно перегляду сцени; 6 – панель інспектора обраного об'єкта;
- 7 – кнопка запуску режиму симуляції.

- створити нову сцену;
- побудувати полігональну модель, наприклад, створивши базові примітиви та застосувавши до них модифікатори;
- додати, розмістити на сцені та налаштувати джерела світла, камеру;
- додати об'єкти, які визначатимуть межі кімнат – їм необхідно призначити компонент *RoomBounds* та задати розміри;
- додати об'єкти, які визначатимуть межі перегородок – їм слід призначити компонент *PartitionBounds*, задати розміри, матеріал та вибрати з переліку кімнати, до яких вона належить;
- додати розмістити на сцені джерела звуку – даним об'єктам сцени необхідно призначити компонент *AudioSourceTarget*;
- налаштувати компонент *AudioSource* джерел звуку – обрати ресурсний файл аудіо з переліку для поля *AudioClip*, встановити прапорець *Spatialize* та задати параметру *SpatialBlend* значення 1 (рис. 4.2 е);
- додати слухача на сцену – йому слід призначити компонент *AudioListenerTarget* (для зручності цей компонент призначають об'єкту, якому належить компонент камери).

Для точного позиціювання об'єктів слід використовувати інструмент сіток у вікні перегляду сцени або вказувати значення в інспекторі.

Вказати позицію та розміри кімнат та перегородок можливо як у вікні перегляду сцени з допомогою допоміжних маніпуляторів, так і введенням відповідних значень на панелі інспектора (рис. 4.2 б-д).

За необхідності користувачі визначення матеріалів перегородок додаються до директорії *PartitionMaterials* з панелі файлової системи проєкту за допомогою пункту *PartitionMaterialProperties* контекстного меню. Налаштувати необхідні октавні коефіцієнти для створеного матеріалу можна на панелі інспектора (рис. 4.2 а).

Для запуску симуляції оточення слід натиснути відповідну кнопку-перемикач у верхній частині вікна редактора.

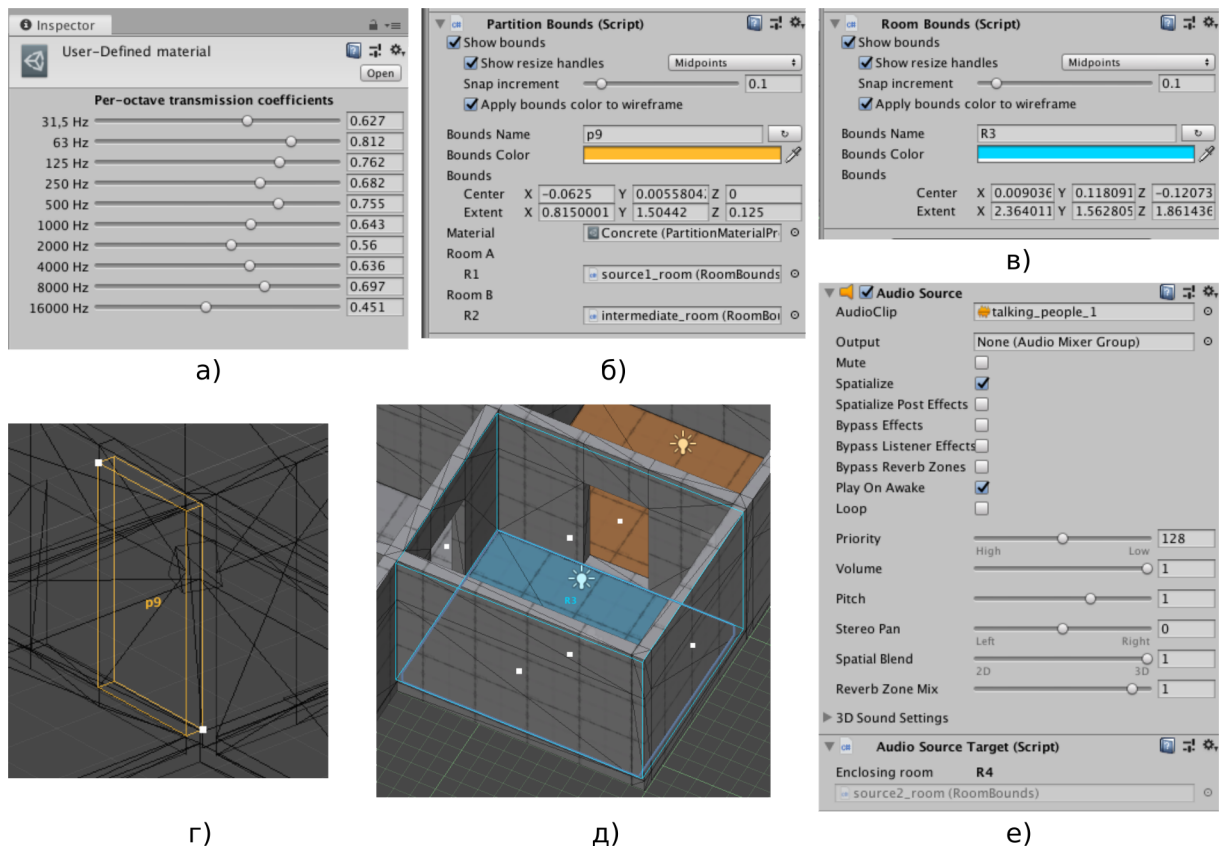


Рис. 4.2. Інтеграція розширення в інтерфейсі редактора Unity:

а) налаштування коефіцієнтів матеріалу перегородок; б-в) параметри перегородки та кімнати в інспекторі; г-д) допоміжні маніпулятори для перегородки та кімнати у вікні сцени; е) налаштування джерела звуку.

4.4. Створення тестового набору акустичних подій

Набір попередньо записаного аудіо було створено на основі даних з публічно доступної бази аудіозаписів Freesound [26] та музичних творів Кевіна МакЛауда [27] розповсюджуваних на умовах ліцензії Creative Commons. Обрані аудіозаписи відповідають типовим шумам в будівлях – голосні розмови, плач немовля, шум води в душі, увімкнене радіо та музика.

В процесі підготовки всі аудіофайли були зведені до монофонічних. До записів з Freesound додатково застосовано редагування та нормалізацію. Здебільшого обиралися оригінали з мінімальною реверберацією та фоновим шумом.

4.5. Аналіз отриманих результатів

Прослуховування отриманого аудіосигналу показало, що необхідно додатково враховувати явище дифракції при обчисленні графа передатних функцій шляху поширення звуку. Крім того, використання HRTF на основі вільного набору KEMAR не у всіх випадках гарантує точну локалізацію джерел звуку. Для виправлення цього слід використовувати інші HRTF, які значно точніше відповідають HRTF слухача.

Аналіз взаємодії користувача з розробленим продуктом виявив необхідність у допоміжній візуалізації у вікні огляду сцени шляхів поширення та зв'язків кімнат з перегородками, засобах експорту змодельованої сцени для спрощення спільної роботи над нею, а також впровадження використання заготовок (так званих *prefab*) – шаблонних об'єктів для сцени.

4.6. Висновки до розділу 4

Для забезпечення відповідності кінцевого продукту вимогам, сформульованим в попередньому розділі, було розроблено методику тестування, відповідно до якої продукт слід виконувати методом напівавтоматизованого *Gray Box Testing*. Наведено основні тестові сценарії, які використовуються для ручного тестування.

Було створено тестовий набір акустичних подій на основі публічно доступних аудіозаписів та побудовано тестову модель набору приміщень. Виконано аналіз отриманих результатів та встановлено напрямки вдосконалення.

5. СТАРТАП

5.1. Аналіз проблеми

Офісні та житлові приміщення в більшості випадків розміщені поряд з джерелами шуму різних типів – сусідніми приміщеннями, ланками транспортної мережі, промисловими підприємствами, стадіонами, тощо. Тому необхідність оцінювати ефективність шумоізоляції є особливо актуальною на етапі їх проектування або реконструкції.

Ступінь впливу шуму на людину залежить від його рівня, характеру та тривалості впливу, а також індивідуальних особливостей людини. Основний вплив шуму доводиться на слуховий апарат людини. Тривала дія шуму високої інтенсивності може поступово призвести до поступової втрати слуху – починаючи з високих частот і поступовим поширенням на більш низькі.

Окрім безпосереднього впливу на орган слуху, шум може змінювати процеси вищої нервової діяльності, призводячи до підвищеної втомлюваності, зниженої працездатності, роздратованості, апатії, послаблення уваги та пам'яті, тощо. Таким чином, шум сприяє зниженню продуктивності та виникненню нещасних випадків.

Несприятлива дія навіть незначних за рівнем шумів сильніше проявляється у години відпочинку. Тому, згідно з законом України “Про забезпечення санітарного та епідемічного благополуччя населення” [1], у житлових масивах діє заборона порушення тиші з двадцять другої до восьмої години, а шум при здійсненні будь-якої діяльності не повинен перевищувати рівнів, встановлених санітарними нормами для відповідного часу доби. Крім того, “проведення ремонтних робіт, що супроводжуються шумом, забороняється у робочі дні з двадцять першої до восьмої години, а у святкові та неробочі дні – цілодобово”. Окремим винятком є лише згода мешканців усіх прилеглих квартир.

Недостатня звукоізоляція, окрім проблем зі здоров'ям, може бути причиною порушення приватності – власник приміщення може отримувати небажану для нього інформацію від сусідів або навпаки. Щонайменше це призводить до відсутності комфорту та напруженості при перебуванні у приміщенні, в тому числі – через потенційну можливість поширення пліток, витоку конфіденційної інформації. І вдосконалення звукоізоляції власник приміщення буде змушений проводити за власні кошти.

Крім того, недостатня звукоізоляція може значно знижувати вартість приміщення, що не дасть можливості забудовнику або власнику вигідно його продати.

Оптимальний вибір звукоізоляційних матеріалів та їх розміщення дозволяє досягнути відповідності нормам рівня шуму з мінімальними витратами. Для досягнення цієї мети дизайнери та проєктувальники під час розгляду проєкту приміщення стикаються з необхідністю виконувати обчислення вручну, що є трудомістким процесом та мають високий ризик допустити помилку при розрахунку. Водночас автоматизовані рішення не надають можливості аудіально, тобто на слух, оцінити наскільки комфортну для власника звукоізоляцію дадуть обрані матеріали. Крім того, можуть не враховуватися побажання клієнта. Наприклад, встановлена вздовж стіни шафа може знизити кількість необхідного для ізоляції матеріалу, а стіна, яка розташована поряд з робочим місцем, навпаки, потребуватиме додаткову ізоляцію.

Наведені вище проблеми подано ієрархічно на рис. 5.1.

5.2. Зацікавлені сторони

Аналіз проблеми дозволяє виявити наступні зацікавлені сторони: власник приміщення, проєктувальник, дизайнер, забудовник та виробник будівельних і звукоізоляційних матеріалів.

Забудовник в більшості випадків зацікавлений у максимізації прибутку і нерідко шляхом економії на матеріалах, що не завжди сприяє



Рис. 5.1. Дерево проблем

достатній звукоізоляції. Проте відповідність приміщень встановленим нормам може бути однією з їх конкурентних переваг.

Власник приміщення зацікавлений у досягненні прийнятного рівня шуму в приміщенні під час його експлуатації, вдосконаленні звукоізоляції з мінімальними витратами та врахуванням кінцевого інтер'єру.

Проектувальники мають необхідність в оптимізації процесу оцінки відповідності приміщення нормам через автоматизацію розрахунків (в тому числі – рівня шуму), проте можливість оцінити достатність звукоізоляції на слух для них значно менш важлива, ніж для дизайнерів.

Дизайнери в більшості випадків не мають достатніх знань та навичок для виконання обчислень вручну і віддають перевагу більш інтуїтивним та наочним способам оцінки впливу обраних матеріалів на зміну рівня шуму у приміщенні. Тому спрощення взаємодії дизайнерів та проектувальників з власниками приміщень є найбільш пріоритетною задачею і саме на ці групи осіб орієнтований створюваний програмний продукт. При цьому

максимально впливовою зацікавленою стороною будуть проєктувальники та дизайнери.

Виробники будівельних та звукоізоляційних матеріалів можуть бути зацікавленими в наявності матеріалів, які вони виробляють, в базі даних матеріалів розроблюваного програмного продукту. Це дозволить його користувачам більш точно моделювати оточення. З даною зацікавленою стороною можливе створення взаємовигідного партнерства.

Для привернення уваги сторін знадобиться участь у спеціалізованих виставках, демонстрація можливостей програмного продукту та його цінності для кожної зі сторін. Для дизайнерів додатково слід проводити майстер-класи та навчання на реальних прикладах, в тому числі – безкоштовні.

В табл. 5.1 зведена загальна інформація про зацікавлені сторони.

Таблиця 5.1

Зацікавлені сторони

Зацікавлена сторона	Інтерес	Вплив	Стратегії приваблення для сторони
Забудовник	Відповідність приміщень встановленим нормам, мінімізація вартості проєктування	низький	Участь у спеціалізованих виставках, демонстрація можливостей
Проєктувальник	Простота оцінки приміщень на відповідність нормам, оптимізація процесу проєктування	середній	
Власник приміщення	Прийнятний рівень шуму у приміщенні	високий	
Дизайнер	Можливість оцінювати рівень шуму максимально просто і інтуїтивно	середній	Проведення майстер-класів, навчання

Зацікавлена сторона	Інтерес	Вплив	Стратегії приваблення для сторони
Виробники звукоізоляційних та будівельних матеріалів	Наявність їх матеріалів у базі даних програмного продукту	низький	Участь у спеціалізованих виставках

5.3. Комерційне рішення. Основні характеристики

Вирішення означених у підрозділі 5.1 проблем потребує створення програмного продукту, в основі реалізації якого лежить описаний в попередніх розділах спосіб ауралізації поширення повітряного шуму в будівлях. Спосіб надає можливість ефективно відтворювати шум у вказаному місці цільового приміщення, використовуючи інформацію про обрані для стін приміщення будівельні та звукоізоляційні матеріали та наявні зовнішні приміщення джерела шуму.

Даний програмний продукт призначений для органічної інтеграції в робочі процеси проектування та реконструкції будівель. В загальному випадку процесі проектування можливо виділити чотири етапи [28]:

- передпроектний – складання техзавдання, місце майбутньої забудови всебічно обстежується, створюються ескізи для спрощення прийняття рішень щодо подальшого проектування, обумовлюються поставлені цілі та завдання;
- оформлення договору після успішного завершення першого етапу процесу проектування;
- складання проєкту – узгодження отриманих схем із замовником та іншими підрозділами, які беруть участь у спорудженні об'єкта, після чого проводиться оптимізація вартості;
- проведення експертизи та затвердження погоджених проєкту та кошторису на проведення будівництва.

Програмний продукт надаватиме можливість оцінювати найбільш інтуїтивним способом – на слух – зміну шуму в залежності від обраних матеріалів перегородок, перекриттів та додаткових конструкцій.

Доповнення наявних автоматизованих засобів розрахунку відповідності нормам рівня шуму в приміщенні таким чином знижуватиме ймовірність виникнення помилки в обчисленнях. При цьому програмний продукт не вимагатиме від користувачів значного обсягу професійних знань в галузі будівельної акустики та дозволить наочно продемонструвати потенційні результати забудовнику чи власнику приміщення.

Очевидно, що основними клієнтами даного програмного продукту будуть проєктувальники та в значно меншій мірі – дизайнери, тому взаємодія з ними побудована на моделі B2B, тобто “бізнес для бізнесу”.

5.4. Конкурентні переваги рішення

Результати тривалих досліджень впливу шуму на людину та розвиток галузі структурної акустики призвели до появи цілого спектру програмних та апаратних рішень для прогнозування та вимірювання рівня шуму в середовищі, моделювання акустики приміщень.

Основним конкурентним програмним рішенням, здатним виконувати ауралізацію поширення шуму всередину цільового приміщення, є BASTIAN компанії DataKustic [29]. Всі обчислення виконуються відповідно до серії стандартів ISO 12354. При моделюванні оточення користувач вказує розташування цільового приміщення та приміщення-джерела, типи стиків та матеріали перегородок між приміщеннями, а за необхідності – додаткові шляхи передачі шуму – вікна, двері або вентиляційні та кабельні канали. Типи матеріалів перегородок входять до бібліотек матеріалів, які, окрім базової, користувачі мають купувати окремо. Крім того, для моделювання поширення шуму з-поза меж будівлі програмне забезпечення інтегрується з іншим продуктом цієї компанії, призначеним для моделювання розповсюдження шуму на

відкритому просторі (наприклад, на вулицях міста) – CadnaA [30]. Мінімальний комплект програмного забезпечення BASTIAN – сама програма та базова бібліотека матеріалів коштуватимуть 1860 євро, із додатковими трьома бібліотеками матеріалів ціна зростає до 2360 євро. Опція ауралізації вимагатиме додаткових 840 євро [31].

В більшості випадків програмні продукти, які надають можливість ауралізації, призначені для архітектурної акустики, тобто виконують симуляцію поширення звукових хвиль в межах приміщення, віддзеркалення та поглинання їх поверхнями. До такого програмного забезпечення належать, наприклад, Odeon та CadnaR [32, 33].

Програмними продуктами, які можуть надавати схожі технологічні засоби для відтворення поширення звуку у будівлях та на відкритому просторі, є також продукти для віртуальної реальності. Наприклад, таким є дослідницький проєкт Virtual Acoustics [34] Рейнсько-Вестфальського технічного університету Аахена (RWTH Aachen University). Донедавна він був доступний лише для досліджень в стінах університету, проте наразі частина його вихідного коду опубліковано на умовах вільних ліцензій.

5.5. Клієнти. Сегменти ринку споживання

У попередніх підрозділах було визначено, що основними клієнтами даного програмного продукту є проєктувальники. Основна їх задача – виконувати проєктування архітектурних об'єктів відповідно до основних вимог надійності та безпеки, серед яких – забезпечення механічного опору та стійкості, дотримання вимог пожежної безпеки, забезпечення безпеки життя і здоров'я людини, захисту навколишнього середовища, забезпечення безпеки експлуатації, захисту від шуму та економії енергії.

На сьогодні відбувається активне будівництво нових споруд і наявний помітний попит на послуги з проєктування будівель. На даному ринку споживання присутня незначна сегментація за розміром компанії. Крім того, можливо сегментувати ринок за спеціалізацією: промислові

об'єкти, житлові та громадські будівлі, об'єкти інфраструктури, заклади освіти та охорони здоров'я тощо.

Потенційними клієнтами є середні та великі компанії, оскільки для більшості малих компаній використання даного програмного продукту підвищуватиме вартість та складність проєкту і не є рентабельним. За спеціалізацією найбільш перспективними є компанії, що проєктують промислові об'єкти, заклади охорони здоров'я та освіти, житло та офіси.

5.6. Унікальна ціннісна пропозиція

Ціннісна пропозиція – твердження, яке визначає чіткі, вимірні та наочні переваги, які отримують споживачі при купівлі певного товару або послуги. Воно має переконувати споживача в тому, що даний товар чи послуга є кращими за інші доступні на ринку у розв'язанні проблем споживача.

Попередньо були викладені результати аналізу проблем користувачів, відповідно до яких виявлені основні зацікавлені сторони та потенційні клієнти – компанії, що надають послуги проєктування та реконструкції будівель. Отже, унікальну ціннісну пропозицію можна сформулювати наступним чином: “Для компаній, що виконують проєктування будівель, даний програмний продукт дозволить оцінювати ефективність використаних шумоізоляційних матеріалів найбільш інтуїтивним способом – на слух. На відміну від програмних рішень, які подають інформацію про ефективність звукоізоляції у вигляді числових показників та графічно”.

5.7. Доходи та витрати

Основним джерелом доходу є продаж поновлюваних ліцензій на використання програмного забезпечення, бібліотек властивостей будівельних та звукоізоляційних матеріалів, а також надання розширеної технічної підтримки. Під поновлюваністю мається на увазі надання ліцензії

на тимчасове користування послугою або програмним продуктом, тобто використовується так звана модель підписки. Платна розширена технічна підтримка, у порівнянні зі стандартною, скорочує терміни надання відповіді на запити користувача та усунення дефектів. Іншими додатковими послугами є адаптація програмного забезпечення та розширення функціональності в залежності від потреби користувача, створення додаткових бібліотек матеріалів.

Основні статті витрат складатимуть:

- проведення науково-дослідних та дослідно-конструкторських робіт;
- оплата праці персоналу;
- утримання робочих місць персоналу та виробничої інфраструктури;
- придбання обладнання, ліцензій програмного забезпечення для розробки;
- податкові витрати.

Витрати на реалізацію проєкту та прогнозовані прибутки за перший рік діяльності стартап-проєкту зведені у табл. 5.2.

Таблиця 5.2

Витрати на реалізацію проєкту, у тис. доларів США

Найменування витрат	Місяці						
	1	2	3	4	5	6	7
Загальні витрати	6	4	4	4	4	4	4
ЗП	2	10	10	10	10	10	10
Підсумок витрат	8	14	14	14	14	14	14
Заплановані доходи	—	—	—	—	—	—	—
Прибуток (без оподаткування)	-8	-14	-14	-14	-14	-14	-14

Найменування витрат	Місяці					Загальні результати
	8	9	10	11	12	
Загальні витрати	4	4	4	4	4	50
ЗП	10	10	10	10	10	112
Підсумок витрат	14	14	14	14	14	162
Заплановані доходи	40	50	50	50	55	245
Прибуток (без оподаткування)	26	36	36	36	41	83

Про комерційну життєздатність даного стартап-проекту можливо стверджувати не зважаючи на те, що проведений аналіз не враховує всіх факторів та ризиків, пов'язаних, наприклад, із податковою політикою країни, в якій ведеться бізнес.

5.8. Бізнес-модель

В даному підрозділі слід узагальнити попередні підрозділи у вигляді бізнес-моделі, побудованої за шаблоном lean canvas, яка у зведеному вигляді наведена в табл. 5.3.

Споживачі: компанії, що виконують проєктування будівель та дизайнери.

Проблема: відсутність або низька наочність автоматизованих рішень; висока вартість людських послуг; людський фактор.

Рішення: програмне забезпечення, що дозволяє відтворювати звукове поле в приміщеннях, створюване зовнішніми джерелами шуму.

Унікальна ціннісна пропозиція: можливість оцінювати ефективність шумоізоляції приміщення для найбільш інтуїтивним способом – на слух.

Канали: відділи інтеграції компаній, які займаються проєктуванням та реконструкцією будівель.

Канва бізнес-моделі

Проблема	Рішення	Унікальна ціннісна пропозиція	Прихована перевага	Споживачі
відсутня або низька наочність автоматизованих рішень	програмний продукт, здатний відтворювати поширення повітряного шуму в межах множини приміщень	оцінка ефективності шумоізоляції приміщення для найбільш інтуїтивним способом – на слух	органічна інтеграція в робочі процеси проєктування та реконструкції будівель	компанії, що виконують проєктування та реконструкцію будівель
висока вартість людських послуг				
людський фактор	Ключові показники		Канали	
	кількість проданих ліцензій		відділи інтеграції компаній-споживачів	
Структура витрат		Потоки доходів		
<p>проведення науково-дослідних та дослідно-конструкторських робіт;</p> <p>оплата праці персоналу</p> <p>утримання робочих місць персоналу та виробничої інфраструктури</p> <p>обладнання, ліцензій програмного забезпечення</p> <p>податкові витрати</p>		<p>доходи від продажу ліцензій програмного забезпечення та бібліотек матеріалів</p> <p>доходи від підтримки програмного забезпечення</p>		

Потоки доходів: доходи від продажу ліцензій програмного забезпечення та бібліотек матеріалів; доходи від підтримки програмного забезпечення.

Ключові метрики: кількість проданих ліцензій.

Структура витрат: проведення науково-дослідних та дослідно-конструкторських робіт; оплата праці персоналу; утримання робочих місць персоналу та виробничої інфраструктури; придбання обладнання, ліцензій програмного забезпечення для розробки; податкові витрати.

Прихована перевага: органічна інтеграція в робочі процеси проєктування та реконструкції будівель; наочність.

Про перспективність реалізації програмного продукту на основі розробленого способу ауралізації можливо стверджувати виходячи з актуальності проблем потенційних клієнтів, які можливо вирішити, наявності конкурентних переваг запропонованого рішення та аналізу комерційної життєздатності проєкту.

5.9. Висновки до розділу 5

У даному розділі з метою дослідження комерційної життєздатності програмної реалізації розробленого способу було розглянуто можливість створення стартап-проєкту на її основі.

В першу чергу було проведено аналіз важливості ефективної звукоізоляції, виявлення проблем, які виникають у зв'язку з її недостатньою ефективністю, а також причин які цьому передують. Ієрархічний узагальнений перелік проблем подано у вигляді дерева проблем. Відповідно до означених проблем виявлені основні зацікавлені сторони, їх інтереси та вплив, а також стратегії приваблення.

Надалі було запропоновано комерційне рішення, в основі якого лежить розроблюваний спосіб ауралізації поширення повітряного шуму, розглянуто його конкурентні переваги.

На основі аналізу потенційних клієнтів та сегментів ринку споживання сформульовано унікальну ціннісну пропозицію та зроблено прогнозування потенційних доходів та витрат на реалізацію стартап-проєкту.

Наведена бізнес-модель узагальнює всі проведені дослідження та аналіз доцільності реалізації даного програмного продукту.

ВИСНОВКИ

Метою даної магістерської дисертації було розроблення способу ауралізації поширення повітряного шуму від джерел до слухача, які розміщені в межах множини кімнат.

На початковому етапі роботи над дисертацією було виконано наступні роботи:

- аналіз загальних підходів та технічних засобів моделювання звукових полів;
- аналіз нормативних документів, які регламентують допустимі рівні шуму в будівлях;
- аналіз теоретичного підґрунтя та математичних моделей акустики, які стосуються поширення звуку в будівлях, а також принципів психоакустики, які враховуються при ауралізації;
- опанування підходів до створення цифрових фільтрів для оброблення аудіосигналу;
- формулювання науково-інноваційної задачі.

На основі результатів виконання цих робіт було сформульовано спосіб ауралізації поширення повітряного шуму в будівлі. Модельоване оточення подається як мультиграф без петель, в якому вузлами є кімнати, а ребрами – конструктивні елементи-перегородки. Пошук шляхів поширення звуку та їх передатні функції дозволяє обчислити імпульсний відгук середовища, який буде застосовано до вихідних сигналів джерел звуку множиною октавних фільтрів.

Було проведено виявлення та аналіз вимог до інноваційного програмного продукту, який реалізує спосіб ауралізації. Розглянуто два варіанти виконання програмного продукту: як розширення для редакторів тривимірних моделей або як окремої прикладної програми, побудованої з використанням програмних рушіїв з підтримкою 3D графіки та систем доповненої реальності. На основі порівняльного аналізу засобів

розроблення, основою для реалізації став рушій Unity. Наведено архітектуру програмного продукту та його ключові алгоритми.

Для забезпечення відповідності розробленого програмного продукту вимогам було розроблено методику тестування, створено тестові моделі оточення та набір акустичних подій. Виконано аналіз отриманих результатів та встановлено напрямки вдосконалення.

Крім того, розглянуто можливість створення стартап-проєкту на основі програмної реалізації розробленого способу. З цією метою проведено аналіз проблем, що виникають через неефективну звукоізоляцію, виявлені основні зацікавлені сторони. Було запропоновано комерційне рішення та розглянуто його конкурентні переваги. Наведена бізнес-модель дозволяє стверджувати про комерційну життєздатність ідеї такого стартап-проєкту.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Про забезпечення санітарного та епідемічного благополуччя населення [Електронний ресурс] // Законодавство України – Режим доступу: <https://zakon.rada.gov.ua/go/4004-12> – Дата доступу: 19.05.2019 – Назва з екрану.
2. Захист територій, будинків і споруд від шуму. ДБН В.1.1-31:2013 [Електронний ресурс] // Державні та галузеві будівельні норми – 2013. – Режим доступу: <http://www.minregion.gov.ua/wp-content/uploads/2017/12/36.1.-DBN-V.1.1-312013.-Zahist-teritoriy-budinkiv-i-sp.pdf> – Дата доступу: 08.03.2019 – Назва з екрану.
3. Санітарні норми виробничого шуму, ультразвуку та інфразвуку. ДСН 3.3.6.037-99 [Електронний ресурс] // Законодавство України – 1999. – Режим доступу: <https://zakon.rada.gov.ua/go/va037282-99> – Дата доступу: 08.03.2019 – Назва з екрану.
4. Kuttruff, H. Room Acoustics [Text] / Heinrich Kuttruff. 5-е вид. – CRC Press, 2009. – 389 p. – ISBN 978-0-203-87637-4.
5. Vorländer, M. Auralization: Fundamentals of Acoustics, Modelling, Simulation, Algorithms and Acoustic Virtual Reality [Text] / Michael Vorländer. – Berlin Heidelberg : Springer-Verlag, 2008. – 335 p. – ISBN 978-3-540-48829-3.
6. American National Standards Institute. S1.11-2004, Specification for Octave-Band and Fractional-Octave-Band Analog and Digital Filters [Text] / American National Standards Institute. // New York: American National Standards Institute – 2004.
7. International Electrotechnical Commission. IEC 61260-1:2014, Electroacoustics – Octave-band and fractional-octave-band filters – Part 1: Specifications [Text] / International Electrotechnical Commission. // IEC – 2014. – Вип. 61260, № 1 – p. 2014.
8. International Organization for Standardization. ISO 226:2003: Acoustics –

- Normal equal-loudness-level contours [Text] / International Organization for Standardization. // International Organization for Standardization, Geneva, Switzerland – 2003. – p. 18.
9. Kleiner, M., Dalenbäck, B.-I., Svensson, P. Auralization – An Overview [Text] / Mendel Kleiner, Bengt-Inge Dalenbäck, Peter Svensson. // J. Audio Eng. Soc – 1993. – Вип. 41, № 11 – pp. 861–875.
10. Savioja, L., Huopaniemi, J., Lokki, T., Väänänen, R. Creating Interactive Virtual Acoustic Environments [Text] / L. Savioja, et al. // Journal of the Audio Engineering Society – 1999. – Вип. 47, № 9 – pp. 675–705.
11. Blauert, Jens. Spatial hearing: the psychophysics of human sound localization [Text] / Jens. Blauert. – Cambridge, Mass. : MIT Press, 1997. – ISBN 0-262-02413-6.
12. Files – Sofaconventions [Электронный ресурс] – Режим доступа: <https://www.sofaconventions.org/mediawiki/index.php/Files> – Дата доступа: 11.11.2019 – Назва з екрану.
13. Kaiser, F. Transaural Audio – The reproduction of binaural signals over loudspeakers [Text] / Fabio Kaiser. – 2011.
14. 7.1 Surround Speakers Setup | Dolby Laboratories [Электронный ресурс] – Режим доступа: [/us/en/guide/surround-sound-speaker-setup/7-1-setup.html](https://www.dolby.com/us/en/guide/surround-sound-speaker-setup/7-1-setup.html) – Дата доступа: 06.10.2019 – Назва з екрану.
15. FIR Filter Properties [Электронный ресурс] // dspGuru – Режим доступа: <https://dspguru.com/dsp/faqs/fir/properties/> – Дата доступа: 02.07.2019 – Назва з екрану.
16. What is Group Delay [Электронный ресурс] – Режим доступа: <http://iowahills.com/B1GroupDelay.html> – Дата доступа: 02.07.2019 – Назва з екрану.
17. Lipshitz, S.P., Pocock, M., Vanderkooy, J. On the Audibility of Midrange Phase Distortion in Audio Systems [Text] / Stanley P. Lipshitz, Mark Pocock, John Vanderkooy. // J. Audio Eng. Soc – 1982. – Вип. 30, № 9 – pp. 580–595.

18. Oppenheim, A.V., Schafer, R.W. Discrete-time signal processing [Text] / Alan V. Oppenheim, Ronald W. Schafer. 3-е вид. – Harlow : Pearson, 2014. – 1052 p. – ISBN 978-1-292-02572-8.
19. Renesas Electronics. AN9603: An Introduction to Digital Filters [Электронный ресурс] / Renesas Electronics. – 1999. – Режим доступа: <https://www.renesas.com/eu/en/doc/application-note/an9603.pdf>.
20. Thede, L. Practical analog and digital filter design [Text] / Les Thede. – Boston : Artech House, 2005. – ISBN 978-1-58053-915-9.
21. Schröder, D. Physically Based Real-Time Auralization of Interactive Virtual Environments [Text] / Dirk Schröder. – Logos Verlag Berlin GmbH, 2011. – 227 p. – ISBN 978-3-8325-3031-0.
22. Architectural Rendering Software | Visualizing Architecture | Autodesk [Электронный ресурс] – Режим доступа: https://www.autodesk.com/solutions/architectural-rendering#architecture_software – Дата доступа: 11.11.2019 – Назва з екрану.
23. 3D Modeling Software Pricing | 3D Design Program Cost [Электронный ресурс] // SketchUp – Режим доступа: <https://www.sketchup.com/plans-and-pricing> – Дата доступа: 11.11.2019 – Назва з екрану.
24. HRTF Measurements of a KEMAR Dummy-Head Microphone [Электронный ресурс] – Режим доступа: <https://sound.media.mit.edu/resources/KEMAR.html> – Дата доступа: 23.10.2019 – Назва з екрану.
25. Unity – Manual: Audio Spatializer SDK [Электронный ресурс] – Режим доступа: <https://docs.unity3d.com/Manual/AudioSpatializerSDK.html> – Дата доступа: 23.10.2019 – Назва з екрану.
26. Freesound – Freesound [Электронный ресурс] – Режим доступа: <https://freesound.org/> – Дата доступа: 04.12.2019 – Назва з екрану.
27. incompetech Music Search [Электронный ресурс] – Режим доступа: <https://incompetech.com/music/royalty-free/music.html> – Дата доступа:

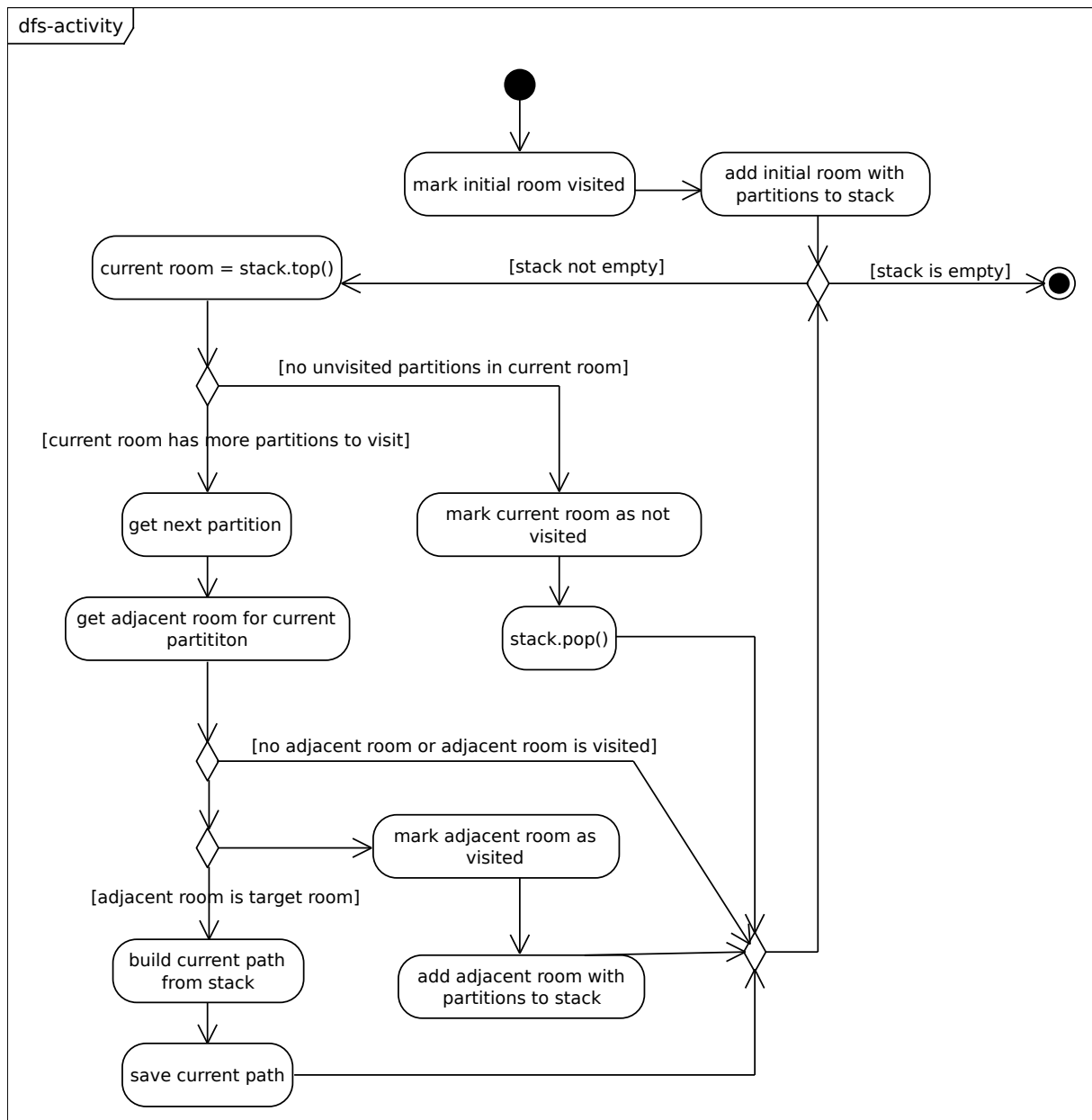
- 04.12.2019 – Назва з екрану.
28. Проектування будівель [Електронний ресурс] – Режим доступу: <http://geospektr.com.ua/ua/proektuvannya-budivel.php> – Дата доступу: 22.05.2019 – Назва з екрану.
29. BASTIAN – The Building Acoustics Planning System [Електронний ресурс] // DataKustik GmbH – 2019. – Режим доступу: <https://www.datakustik.com/products/bastian/bastian/> – Дата доступу: 01.05.2019 – Назва з екрану.
30. CadnaA – State-of-the-art Noise Prediction Software [Електронний ресурс] // DataKustik GmbH – 2019. – Режим доступу: <https://www.datakustik.com/products/cadnaa/cadnaa/> – Дата доступу: 19.05.2019 – Назва з екрану.
31. BASTIAN, prices and condition [Електронний ресурс] – Режим доступу: <https://www.simmons.se/BASTIANprices.html> – Дата доступу: 19.05.2019 – Назва з екрану.
32. The Odeon acoustics software | Simulation and measurements | Odeon.dk // Odeon.
33. CadnaR – Prediction of Sound Inside Rooms [Електронний ресурс] // DataKustik GmbH – 2018. – Режим доступу: <https://www.datakustik.com/products/cadnar/cadnar/> – Дата доступу: 25.05.2019 – Назва з екрану.
34. Virtual Acoustics [Електронний ресурс] – Режим доступу: <http://virtualacoustics.org/> – Дата доступу: 19.05.2019 – Назва з екрану.

ДОДАТКИ

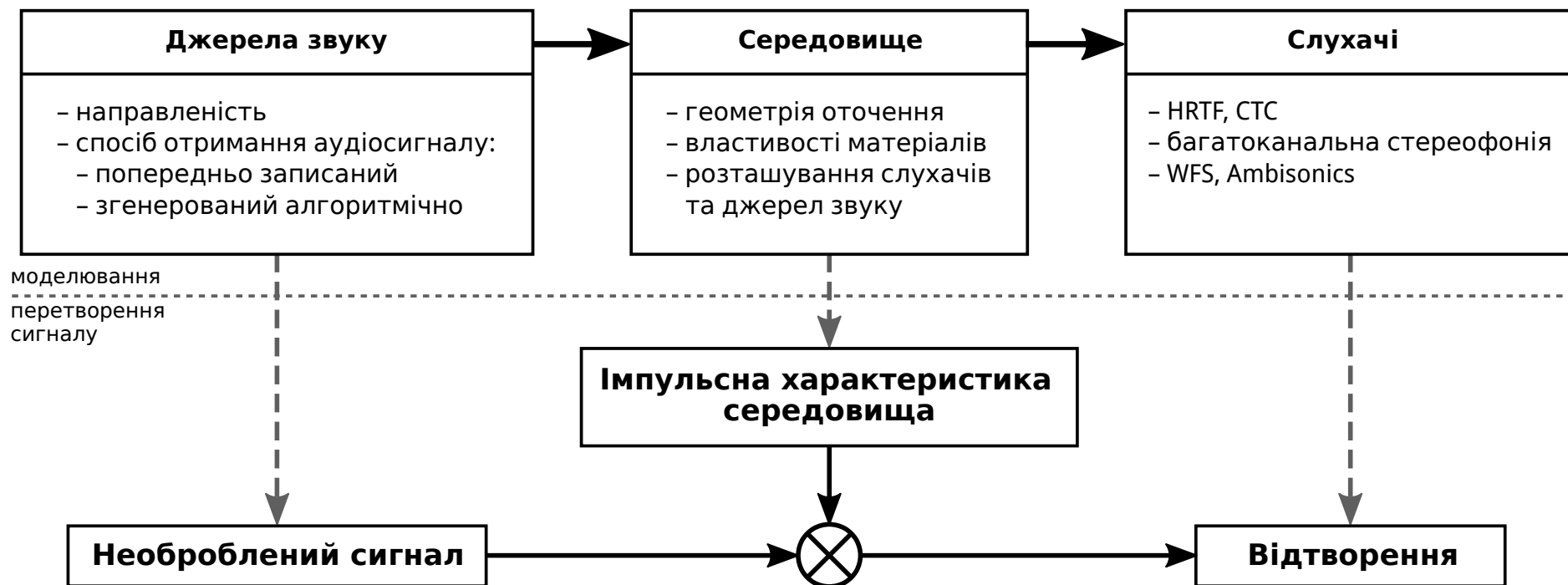
Додаток 1

Копії графічних матеріалів

Діаграма діяльності алгоритму пошуку елементарних шляхів в мультиграфі

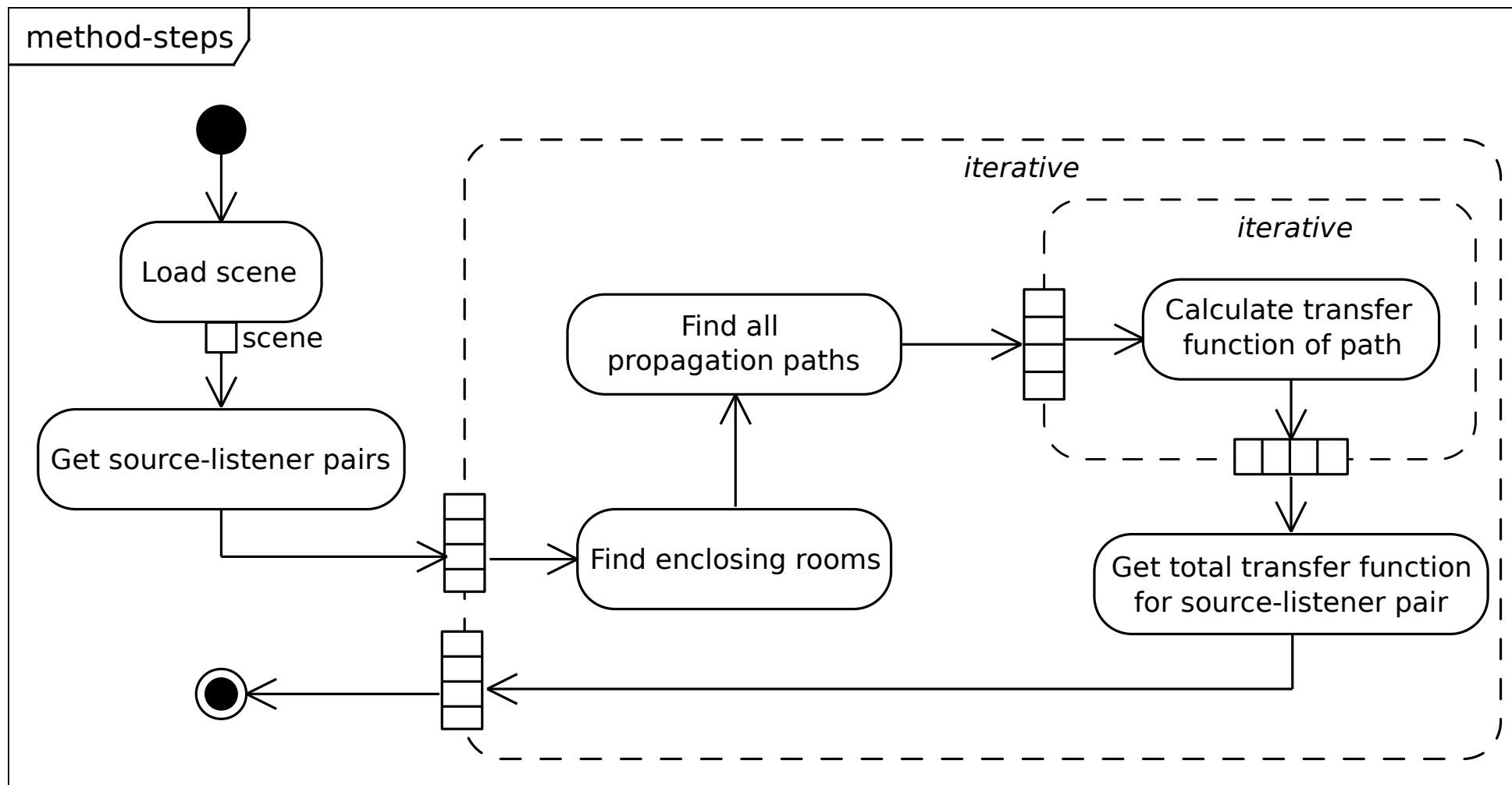


Процеси ауралізації



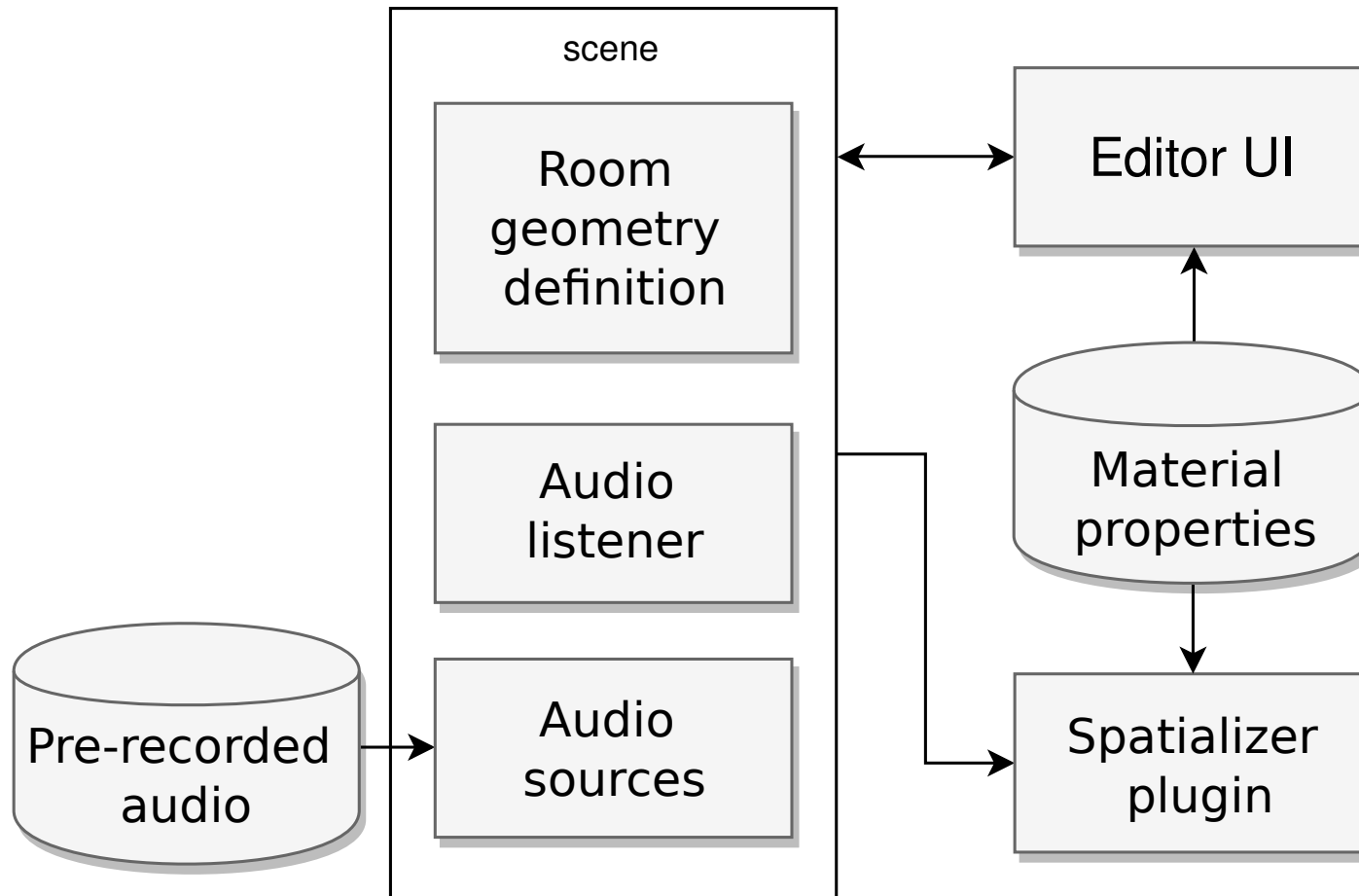
Спосіб та програмне забезпечення ауралізації передачі повітряного шуму
Глінський Владислав Васильович, КП-81мп

Діаграма діяльності кроків способу



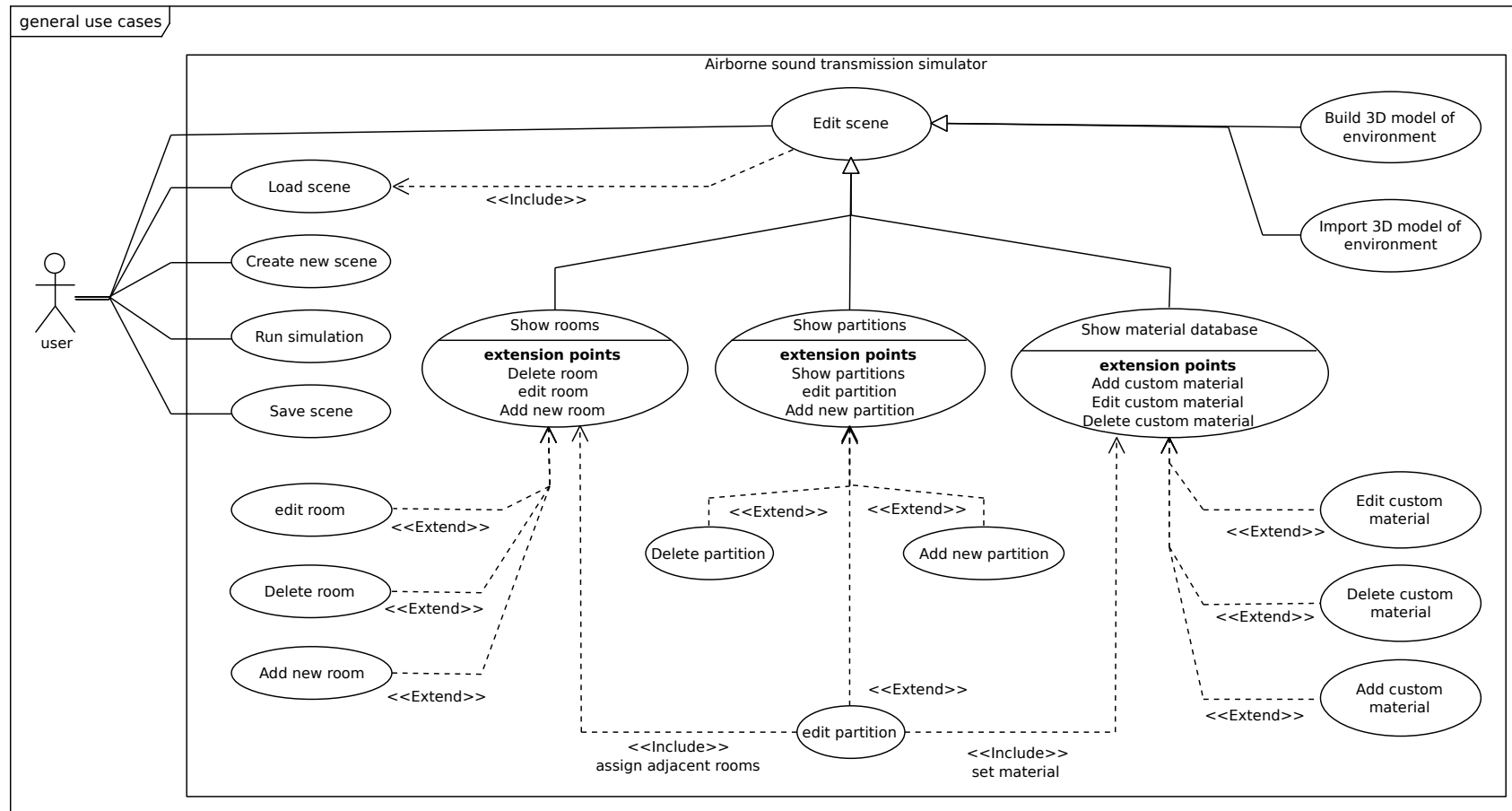
Спосіб та програмне забезпечення ауралізації передачі повітряного шуму
Глінський Владислав Васильович, КП-81мп

Архітектура програмної реалізації способу



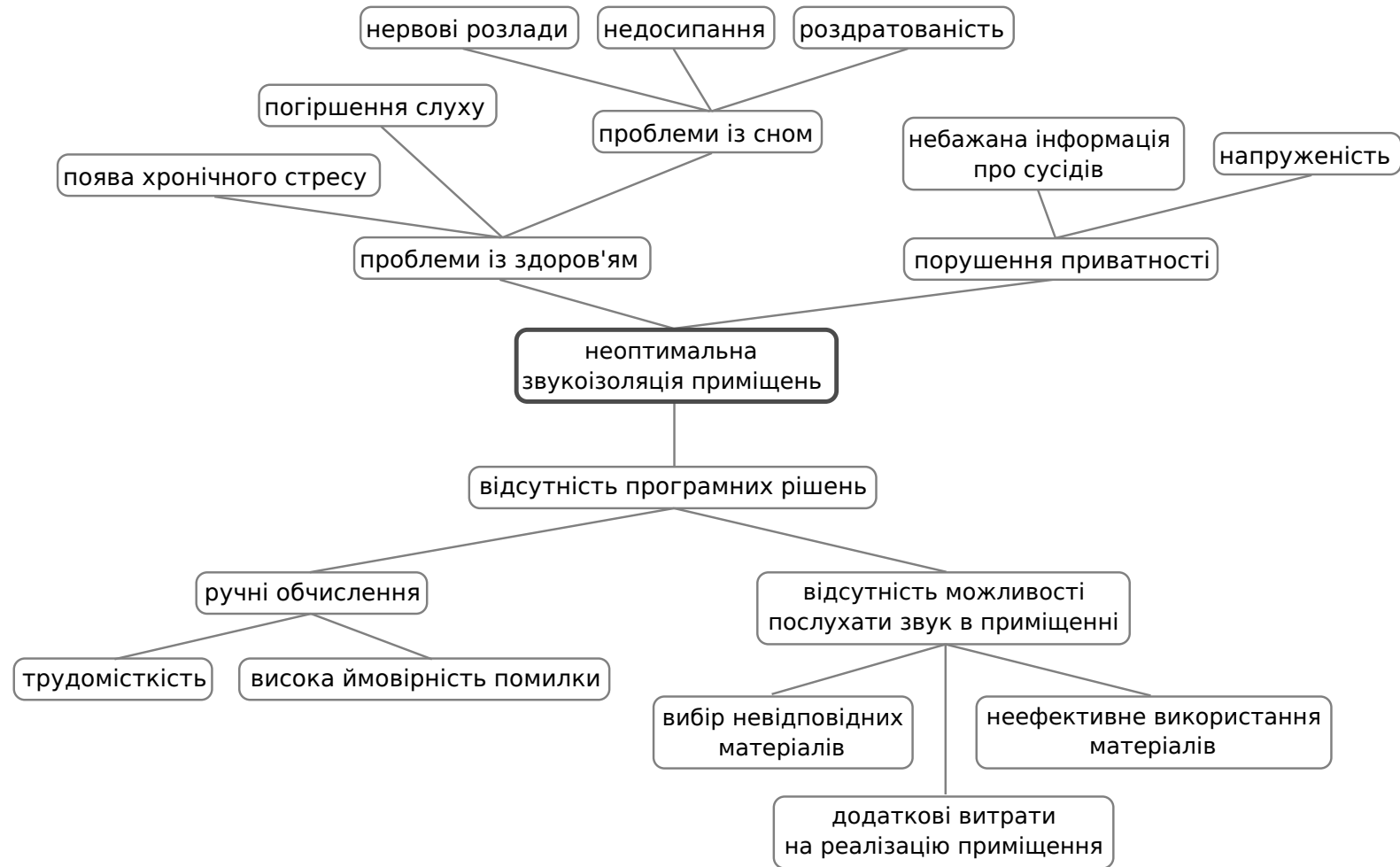
Спосіб та програмне забезпечення ауралізації передачі повітряного шуму
Глінський Владислав Васильович, КП-81мп

Діаграма варіантів використання



Спосіб та програмне забезпечення ауралізації передачі повітряного шуму
Глінський Владислав Васильович, КП-81мп

Дерево проблем



Спосіб та програмне забезпечення ауралізації передачі повітряного шуму
Глінський Владислав Васильович, КП-81мп

Додаток 2

Лістинги коду програмної реалізації способу

Лістинг 1. Модуль реалізації структури даних мультиграф

```
namespace MultiGraphPaths
{
    using System;
    using System.Linq;
    using System.Collections.Generic;

    public class MultiGraph<PV, RV>
    {
        private readonly ISet<Room<PV, RV>> rooms = new HashSet<Room<PV,
RV>>();
        public IEnumerable<Room<PV, RV>> Rooms => rooms;
        public int RoomCount => rooms.Count;

        private readonly ISet<Partition<PV, RV>> partitions = new
HashSet<Partition<PV, RV>>();
        public IEnumerable<Partition<PV, RV>> Partitions => partitions;

        private class RoomPartitionsGetter
        {
            IEnumerator<Partition<PV, RV>> partitions;

            public RoomPartitionsGetter(Room<PV, RV> room)
            {
                Room = room;
                partitions = room.Partitions.GetEnumerator();
            }

            public Room<PV, RV> Room { get; }

            public Partition<PV, RV> NextPartition {
                get {
                    return partitions.MoveNext() ? partitions.Current :
null;
                }
            }

            public Partition<PV, RV> CurrentPartition =>
partitions.Current;
        }

        public bool HasRoom(Room<PV, RV> room)
```

```

    {
        if (room == null) {
            throw new ArgumentNullException(nameof(room));
        }
        return rooms.Contains(room);
    }

    public void AddRoom(Room<PV, RV> room)
    {
        if (room == null) {
            throw new ArgumentNullException(nameof(room));
        }
        rooms.Add(room);
    }

    public void RemoveRoom(Room<PV, RV> room)
    {
        if (room == null) {
            throw new ArgumentNullException(nameof(room));
        }
        rooms.Remove(room);
    }

    public IEnumerable<Path<PV, RV>> GetAllPathsDFS(Room<PV, RV> from,
Room<PV, RV> to)
    {
        if (from == null) {
            throw new ArgumentNullException(nameof(from));
        }
        if (to == null) {
            throw new ArgumentNullException(nameof(to));
        }

        if (from == to) {
            return Enumerable.Empty<Path<PV, RV>>();
        }

        return GetAllPathsDFSImpl(from, to);
    }

    private IEnumerable<Path<PV, RV>> GetAllPathsDFSImpl(Room<PV, RV>
from, Room<PV, RV> to)
    {

```

```

        var visitedRooms = new HashSet<Room<PV, RV>> { from }; // we
can use stack, of course...
        var roomStack = new List<RoomPartitionsGetter> {
            new RoomPartitionsGetter(from)
        };
        RoomPartitionsGetter currentPartitions = null;
        while (roomStack.Any()) {
            if (currentPartitions == null) {
                currentPartitions = roomStack.Last();
            }
            var partition = currentPartitions.NextPartition;
            if (partition == null) {
                visitedRooms.Remove(currentPartitions.Room);
                roomStack.RemoveAt(roomStack.Count-1);
                currentPartitions = null;
                continue;
            }

            Room<PV, RV> adjacent =
partition.GetAdjacentFor(currentPartitions.Room);
            if (visitedRooms.Contains(adjacent)) {
                // we ignore cycles
                continue;
            }

            if (adjacent == null) {
                continue;
            }

            if (adjacent == to) {
                var currentPath = new Path<PV, RV>();
                for (int i = 1; i < roomStack.Count; ++i) {
                    var fromRoom = roomStack[i - 1];
                    var toRoom = roomStack[i];
                    currentPath.Edges.Add(new DirectedEdge<PV, RV> {
                        From = fromRoom.Room,
                        To = toRoom.Room,
                        Value = fromRoom.CurrentPartition.Value
                    });
                }
                currentPath.Edges.Add(new DirectedEdge<PV, RV> {
                    From = currentPartitions.Room,
                    To = adjacent,

```

```

        Value = partition.Value
    });
    yield return currentPath;
    continue;
}

    visitedRooms.Add(adjacent);
    currentPartitions = new RoomPartitionsGetter(adjacent);
    roomStack.Add(currentPartitions);
}
}
}
}
}

```

```

namespace MultiGraphPaths
{
    using System;
    using System.Collections.Generic;

    public class Room<PV, RV>
    {
        readonly ISet<Partition<PV, RV>> partitions = new
HashSet<Partition<PV, RV>>();
        public IEnumerable<Partition<PV, RV>> Partitions => partitions;
        public int PartitionCount => partitions.Count;
        public RV Value { get; set; }

        public void Attach(Partition<PV, RV> partition)
        {
            if (partition == null) {
                throw new ArgumentNullException(nameof(partition));
            }
            partitions.Add(partition);
        }

        public void Detach(Partition<PV, RV> partition)
        {
            if (partition == null) {
                throw new ArgumentNullException(nameof(partition));
            }
            partitions.Remove(partition);
        }
    }
}

```

```

        public bool HasPartition(Partition<PV, RV> partition)
        {
            if (partition == null) {
                throw new ArgumentNullException(nameof(partition));
            }
            return partitions.Contains(partition);
        }
    }
}

```

```

namespace MultiGraphPaths

```

```

{
    using System;
    using System.Linq;
    using System.Collections.Generic;

    public class Path<PV, RV> : IEquatable<Path<PV, RV>>
    {
        public IList<DirectedEdge<PV, RV>> Edges { get; set; }

        public Path()
        {
            Edges = new List<DirectedEdge<PV, RV>>();
        }

        public Path(IEnumerable<DirectedEdge<PV, RV>> edges)
        {
            Edges = new List<DirectedEdge<PV, RV>>(edges);
        }

        public Path(params DirectedEdge<PV, RV>[] edges)
        {
            Edges = new List<DirectedEdge<PV, RV>>(edges);
        }

        public Path<PV, RV> Reversed()
        {
            return new Path<PV, RV>(Edges.Reverse().Select(e =>
e.Reversed()));
        }

        public override bool Equals(object obj)
        {

```

```

        return Equals(obj as Path<PV, RV>);
    }

    public bool Equals(Path<PV, RV> other)
    {
        return other != null && Edges.SequenceEqual(other.Edges);
    }
}

namespace MultiGraphPaths
{
    using System;
    using System.Linq;
    using System.Collections.Generic;

    public class Partition<PV, RV>
    {
        ISet<Room<PV, RV>> rooms = new HashSet<Room<PV, RV>>();
        public int Value { get; set; }
        public IEnumerable<Room<PV, RV>> Rooms => rooms;
        public int RoomCount => rooms.Count;

        public Room<PV, RV> this[Room<PV, RV> r] {
            get => GetAdjacentFor(r);
        }

        public Room<PV, RV> GetAdjacentFor(Room<PV, RV> room)
        {
            if (room == null) {
                throw new ArgumentNullException(nameof(room));
            }
            if (!rooms.Contains(room)) {
                const string message = "Partition doesn't belong to the
specified room";
                throw new ArgumentException(message, nameof(room));
            }
            return rooms.SingleOrDefault(v => v != room);
        }

        public void SetRooms(Room<PV, RV> a, Room<PV, RV> b)
        {
            if (a == b && a != null) {
                const string message = "Partition must belong to two

```

```

different rooms";
        throw new ArgumentException(message);
    }
    var newRooms = new HashSet<Room<PV, RV>>();
    if (a != null) {
        newRooms.Add(a);
    }
    if (b != null) {
        newRooms.Add(b);
    }
    foreach (Room<PV, RV> r in rooms.Except(newRooms)) {
        r.Detach(this);
    }
    foreach (Room<PV, RV> r in newRooms.Except(rooms)) {
        r.Attach(this);
    }
    rooms = newRooms;
}

public bool BelongsTo(Room<PV, RV> room)
{
    if (room == null) {
        throw new ArgumentNullException(nameof(room));
    }
    return rooms.Contains(room);
}
}

namespace MultiGraphPaths
{
    using System;

    public class DirectedEdge<PV, RV> : IEquatable<DirectedEdge<PV, RV>>
    {
        public int Value { get; set; }
        public Room<PV, RV> From { get; set; }
        public Room<PV, RV> To { get; set; }

        public DirectedEdge<PV, RV> Reversed()
        {
            return new DirectedEdge<PV, RV> { From = To, To = From, Value =
Value };

```

```
}

public override bool Equals(object obj)
{
    return Equals(obj as DirectedEdge<PV, RV>);
}

public bool Equals(DirectedEdge<PV, RV> other)
{
    return other != null &&
        Value == other.Value &&
        From == other.From &&
        To == other.To;
}
}
}
```


Лістинг 2. Розширення редактора Unity – налаштування кімнат та перегородок

```
namespace SoundTransmission
{
    using UnityEngine;
    using UnityEditor;

    public abstract class BoundsBaseEditor : Editor
    {
        private static readonly Vector3[] directions = {
            Vector3.right, Vector3.up, Vector3.forward
        };

        static bool showBounds = true;
        static bool showResizeHandles = true;
        static HandlesType handlesType;
        static bool useRoomColor = false;

        private SerializedProperty boundsColor;
        private SerializedProperty boundsName;
        private SerializedProperty boundsValue;
        private static readonly float defaultHandleSize = 0.04f;
        private static Vector3 pointSnap = Vector3.one * 0.1f;

        private static readonly GUIContent nameFromParentContent = new
        GUIContent("\u21BB", "Reset to GameObject's name");
        private static readonly GUILayoutOption miniButtonWidth =
        GUILayout.Width(30f);

        enum HandlesType
        {
            Midpoints,
            TwoCorners
        }

        protected virtual void OnEnable()
        {
            boundsColor = serializedObject.FindProperty("boundsColor");
            boundsName = serializedObject.FindProperty("boundsName");
            boundsValue = serializedObject.FindProperty("bounds");
            HideCurrentToolIfNeeded();
        }
    }
}
```

```

}

private static void HideCurrentToolIfNeeded()
{
    Tools.hidden = showBounds && showResizeHandles;
}

protected virtual void OnDisable()
{
    Tools.hidden = false;
}

public override void OnInspectorGUI()
{
    ShowEditorOptions();

    EditorGUILayout.Space();

    serializedObject.Update();

    ShowPropertiesEditor();

    serializedObject.ApplyModifiedProperties();
}

protected virtual void ShowPropertiesEditor()
{
    using (new EditorGUILayout.HorizontalScope()) {
        EditorGUILayout.PropertyField(boundsName);
        ShowCopyNameFromParentButton();
    }

    EditorGUILayout.PropertyField(boundsColor);
    EditorGUILayout.PropertyField(boundsValue);
}

private void ShowCopyNameFromParentButton()
{
    if (serializedObject.isEditingMultipleObjects)
        return;
    var apply = GUILayout.Button(
        nameFromParentContent,
        EditorStyles.miniButtonRight,

```

```

        miniButtonWidth
    );
    if (apply) {
        boundsName.stringValue =
serializedObject.targetObject.name;
    }
}

private static void ShowEditorOptions()
{
    showBounds = EditorGUILayout.ToggleLeft("Show bounds",
showBounds);
    using (new EditorGUI.IndentLevelScope()) {
        using (new EditorGUILayout.HorizontalScope()) {
            showResizeHandles = EditorGUILayout.ToggleLeft("Show
resize handles", showResizeHandles);
            HideCurrentToolIfNeeded();
            handlesType =
(HandleType)EditorGUILayout.EnumPopup(handlesType);
        }
        var snapFactor = EditorGUILayout.Slider("Snap increment",
pointSnap.x, 0.01f, 1);
        pointSnap = Vector3.one * snapFactor;

        useRoomColor = EditorGUILayout.ToggleLeft("Apply bounds
color to wireframe", useRoomColor);
    }
}

protected void OnSceneGUI()
{
    if (!showBounds)
        return;
    var boundsWrapper = target as BoundsWrapper;
    if (boundsWrapper == null) {
        return;
    }

    var bounds = boundsWrapper.bounds;
    var parent = boundsWrapper.transform;
    if (useRoomColor) {
        Handles.color = boundsWrapper.boundsColor;
    }
}

```

```

        Handles.matrix = Matrix4x4.TRS(parent.position,
parent.localRotation, parent.localScale);
        Handles.DrawWireCube(bounds.center, bounds.size);

        var labelStyle = new GUIStyle(EditorStyles.boldLabel);
        labelStyle.normal.textColor = Handles.color;
        labelStyle.alignment = TextAnchor.MiddleCenter;
        if (boundsWrapper.boundsName.Length > 0) {
            Handles.Label(bounds.center, boundsWrapper.boundsName,
labelStyle);
        }

        Handles.color = Color.white;

        if (!showResizeHandles) return;

        Vector3 newSize = bounds.size;
        Vector3 newCenter = bounds.center;
        switch (handlesType) {
            case HandlesType.Midpoints:
                ResizeMidpoints(bounds, ref newSize, ref newCenter);
                break;
            case HandlesType.TwoCorners:
                ResizeCorners(bounds, ref newSize, ref newCenter);
                break;
            default:
                return;
        }
        var newBounds = new Bounds(newCenter, newSize);
        if (newBounds != bounds) {
            RecordObject(boundsWrapper);
            boundsWrapper.bounds = newBounds;
        }
    }

    protected abstract void RecordObject(BoundsWrapper bounds);

    private static void ResizeMidpoints(Bounds bounds, ref Vector3
newSize, ref Vector3 newCenter)
    {
        Vector3 halfSize = bounds.extents;
        for (int i = 0; i < 3; ++i) {
            Vector3 direction = directions[i];

```

```

        var leftPoint = bounds.center;
        leftPoint[i] -= halfSize[i];
        leftPoint = Handles.Slider(
            leftPoint,
            direction,
            size: HandleUtility.GetHandleSize(leftPoint) *
defaultHandleSize,
            snap: pointSnap[i],
            capFunction: Handles.DotHandleCap
        );

        var rightPoint = bounds.center;
        rightPoint[i] += halfSize[i];
        rightPoint = Handles.Slider(
            rightPoint,
            direction,
            size: HandleUtility.GetHandleSize(rightPoint) *
defaultHandleSize,
            snap: pointSnap[i],
            capFunction: Handles.DotHandleCap
        );

        newSize[i] = rightPoint[i] - leftPoint[i];
        newCenter[i] = Mathf.Lerp(leftPoint[i], rightPoint[i],
0.5f);
    }
}

private static void ResizeCorners(Bounds bounds, ref Vector3
newSize, ref Vector3 newCenter)
{
    var leftPoint = bounds.min;
    leftPoint = Handles.FreeMoveHandle(
        leftPoint,
        rotation: Quaternion.identity,
        size: HandleUtility.GetHandleSize(leftPoint) *
defaultHandleSize,
        snap: pointSnap,
        capFunction: Handles.DotHandleCap
    );

    var rightPoint = bounds.max;

```

```

        rightPoint = Handles.FreeMoveHandle(
            rightPoint,
            rotation: Quaternion.identity,
            size: HandleUtility.GetHandleSize(rightPoint) *
defaultHandleSize,
            snap: pointSnap,
            capFunction: Handles.DotHandleCap
        );

        newSize = rightPoint - leftPoint;
        newCenter = Vector3.Lerp(leftPoint, rightPoint, 0.5f);
    }
}
}

```

```

namespace SoundTransmission
{
    using UnityEngine;
    using UnityEditor;

    [CustomEditor(typeof(RoomBounds))]
    [CanEditMultipleObjects]
    public class RoomBoundsEditor : BoundsBaseEditor
    {
        protected override void RecordObject(BoundsWrapper bounds)
        {
            Undo.RecordObject(bounds, "Change room bounds");
        }
    }
}

```

```

namespace SoundTransmission
{
    using UnityEditor;
    using UnityEngine;

    [CustomEditor(typeof(PartitionBounds))]
    [CanEditMultipleObjects]
    public class PartitionBoundsEditor : BoundsBaseEditor
    {
        private SerializedProperty material;
        private SerializedProperty roomA;
        private SerializedProperty roomB;
    }
}

```

```

protected override void OnEnable()
{
    roomA = serializedObject.FindProperty("roomA");
    roomB = serializedObject.FindProperty("roomB");
    material = serializedObject.FindProperty("material");
    base.OnEnable();
}

protected override void RecordObject(BoundsWrapper bounds)
{
    Undo.RecordObject(bounds, "Change partition bounds");
}

protected override void ShowPropertiesEditor()
{
    base.ShowPropertiesEditor();

    EditorGUILayout.PropertyField(material);
    ShowRoomProperty(roomA);
    ShowRoomProperty(roomB);
}

private bool ShowRoomProperty(SerializedProperty room)
{
    EditorGUILayout.LabelField(room.displayName);
    using (new EditorGUI.IndentLevelScope()) {
        using (new EditorGUILayout.HorizontalScope()) {
            var roomValue = room.objectReferenceValue as
RoomBounds;

            var label = GUIContent.none;
            if (roomValue != null) {
                label = new GUIContent(roomValue.boundsName);
            }
            return EditorGUILayout.PropertyField(room, label);
        }
    }
}
}
}

```

Додаток 3

Копія слайдів презентації

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО”



ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ
КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ

Спосіб та програмне забезпечення ауралізації передачі повітряного шуму

Виконав: Глінський Владислав Васильович

Науковий керівник: к.т.н., доцент Сулема Євгенія Станіславівна

Київ – 2019



Актуальність дослідження

Проблеми неоптимальної звукоізоляції приміщень:

- Негативний вплив шуму на здоров'я та продуктивність людини
- Порушення приватності, витік інформації
- Зниження вартості приміщення

Причини отриманої неоптимальної звукоізоляції:

- Вибір невідповідних матеріалів та нефективне їх використання
- Ручні обчислення = трудомісткість + висока ймовірність помилки
- Неможливо оцінити ефективність звукоізоляції приміщення аудіально на етапі проєктування/реконструкції



Науково-інноваційна задача

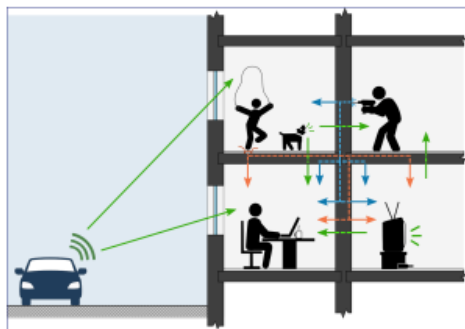
Об'єкт дослідження: процес програмного перетворення аудіосигналу.

Предмет дослідження: способи та алгоритми програмного моделювання акустичного простору.

Наукове завдання: розробити спосіб відтворення характеру зміни звуку в процесі його поширення від джерела до слухача в межах множини приміщень.

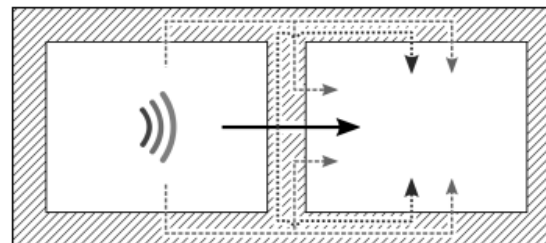
Мета дослідження: створення програмного забезпечення, яке надаватиме засоби аудіальної оцінки ефективності звукоізоляції приміщення на етапі проєктування або реконструкції

Поширення шуму



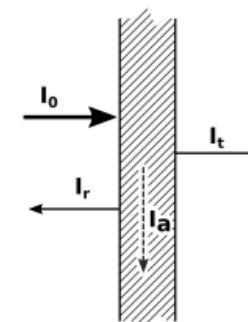
Типи шуму:

- Повітряний
- Структурний
- Ударний



Шляхи поширення:

- Прямі
- Непрямі



Коефіцієнт
звукопроникності
перегородки:

$$\tau = \frac{I_t}{I_0}$$

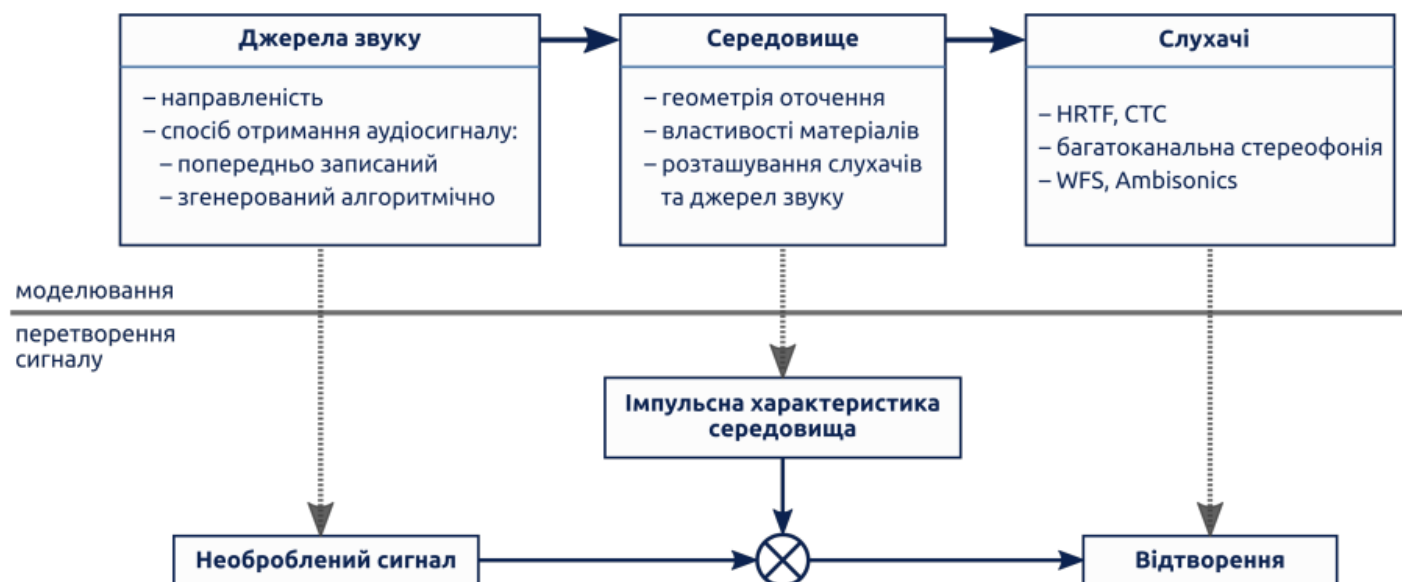
Ауралізація



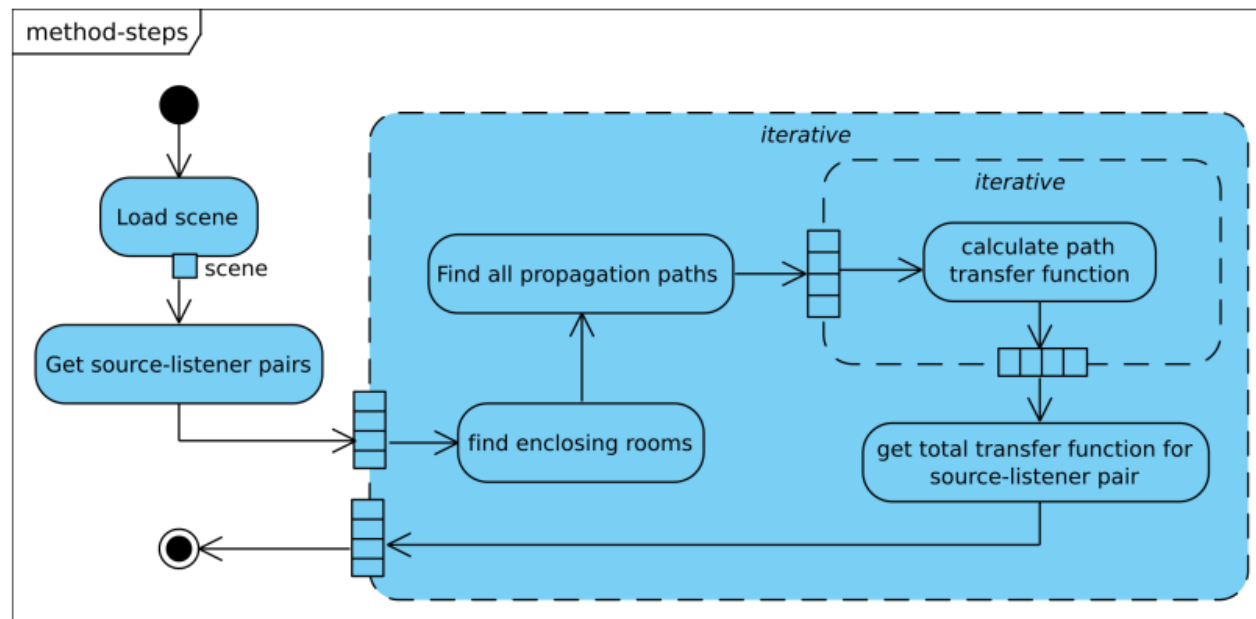
"Auralization is the process of rendering audible, by physical or mathematical modeling, the soundfield of a source in a space, in such a way as to simulate the binaural listening experience at a given position in the modeled space."

Kleiner, Dalenback, Svensson
"Auralization – an overview", 1993

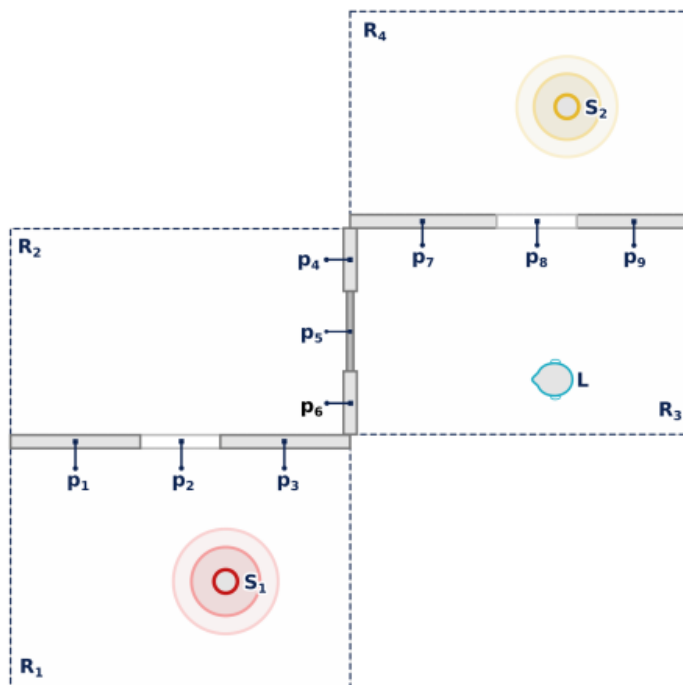
Процес ауралізації



Основні кроки способу



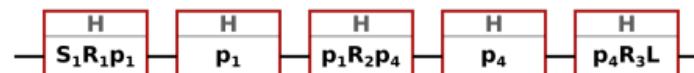
Приклад моделі оточення



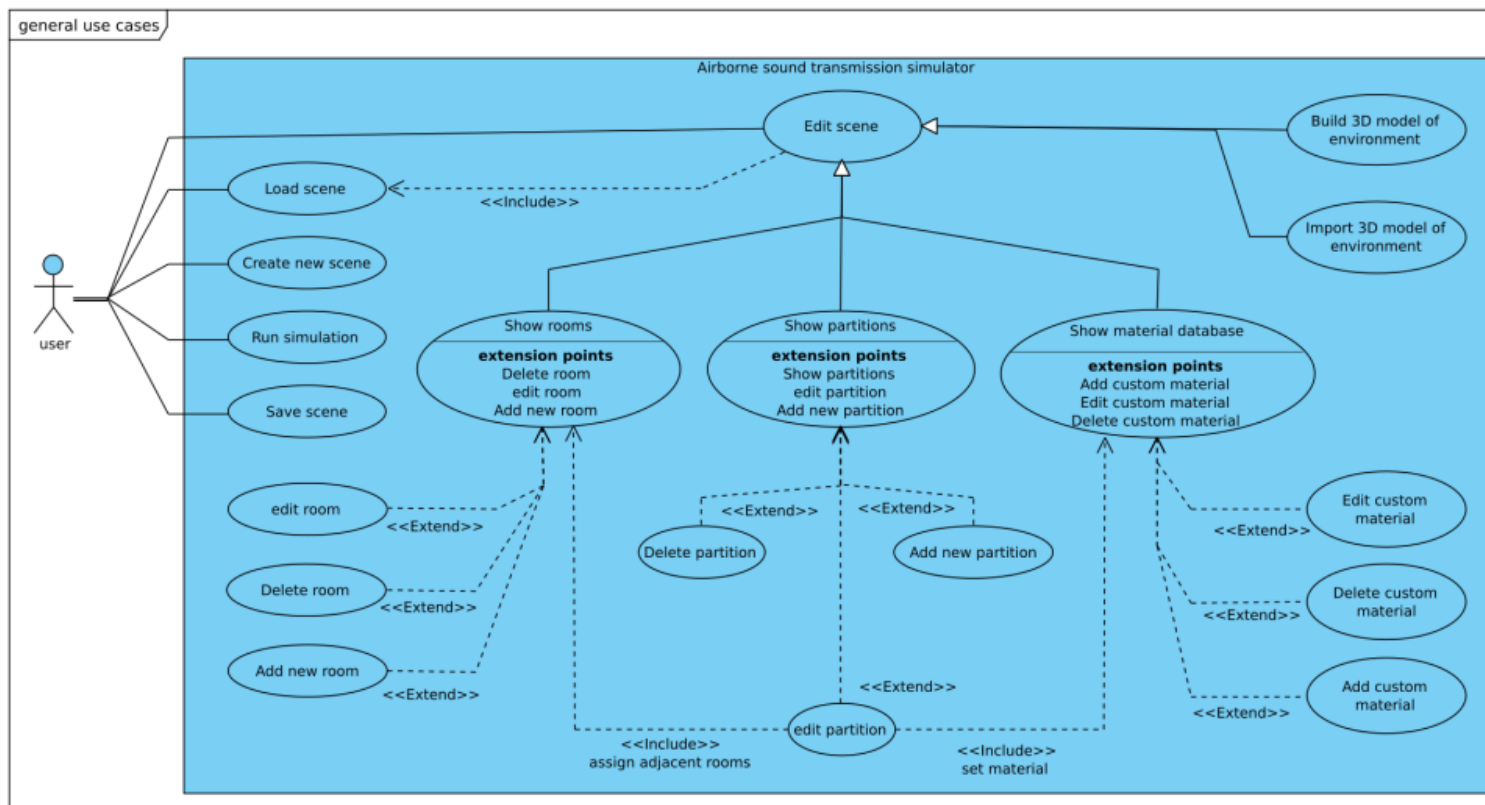
граф зв'язків кімнат



граф передатних функцій одного з шляхів



Вимоги до ПЗ: функціональні



Вимоги до ПЗ: нефункціональні



- графічний інтерфейс
- інтеграція з наявними редакторами тривимірних моделей
- робота в основних ОС для персональних комп'ютерів
- обчислення імпульсного відгуку середовища та оброблення аудіосигналу повинні бути виділені у модуль, що компілюється в машинний код

Вибір засобів розроблення

(1)



редактори тривимірних моделей

		Autodesk 3ds Max, Revit, AutoCAD	SketchUp	Blender
Початкова вартість створення продукту на їх основі		\$1545...2310	\$299	0
Відкрита екосистема розробників		–	+/-	+
Підтримка ОС	Windows	+	+	+
	MacOS X	+/-	+	+
	Linux	–	–	+
Повнота документації для розробників		+/-	+/-	+/-
Простота інтеграції підсистеми аудіо		–	–	–

Вибір засобів розроблення

(2)



програмні рушії

		Unity	UnrealEngine
Початкова вартість створення продукту на їх основі		0	0
Відкритість екосистеми розробників		+	+/-
Підтримка ОС	Windows	+	+
	MacOS X	+	+
	Linux	+	+
Імпорт поширених форматів тривимірних моделей		+	+
Повнота документації для розробників		+	-
Простота інтеграції підсистеми аудіо		+	-



Обрані засоби розроблення

Програмний рушій:

- Unity

Мови програмування:

- C#
- C++

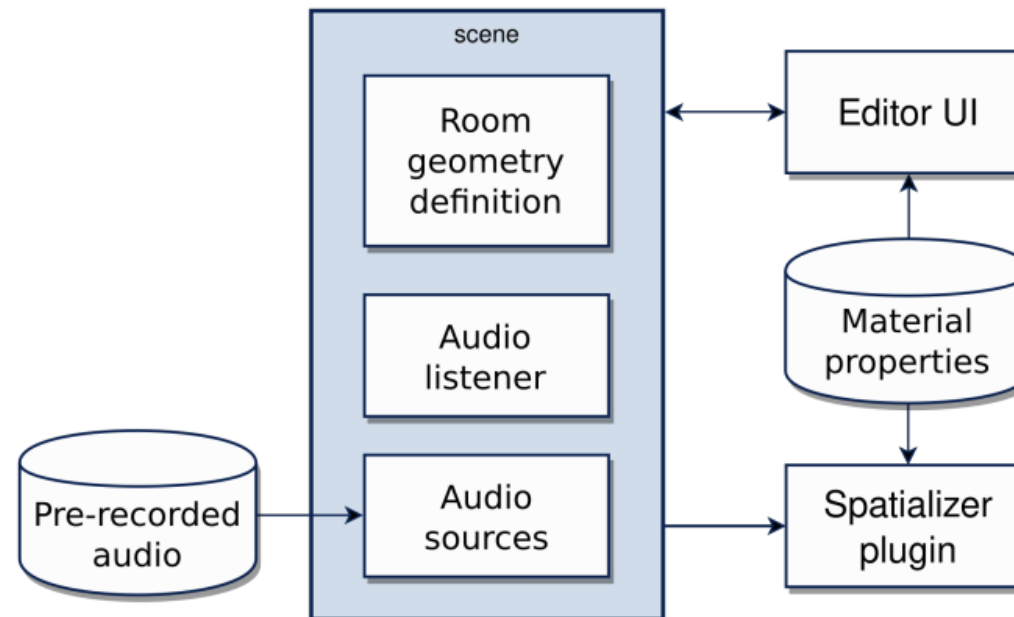
Автоматизація збирання:

- CMake
- MSBuild

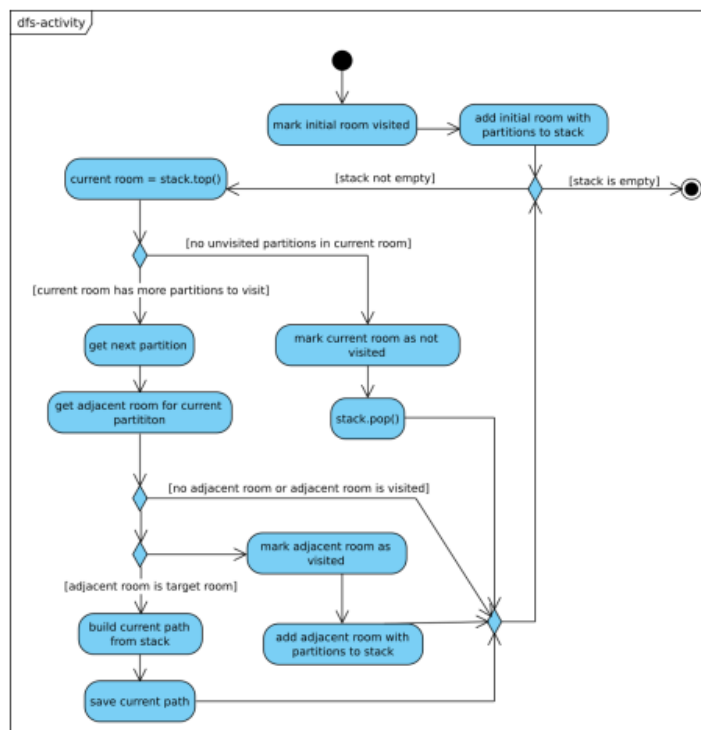
Контроль версій:

- Git

Архітектура розробленого ПЗ



Алгоритм пошуку шляхів



```

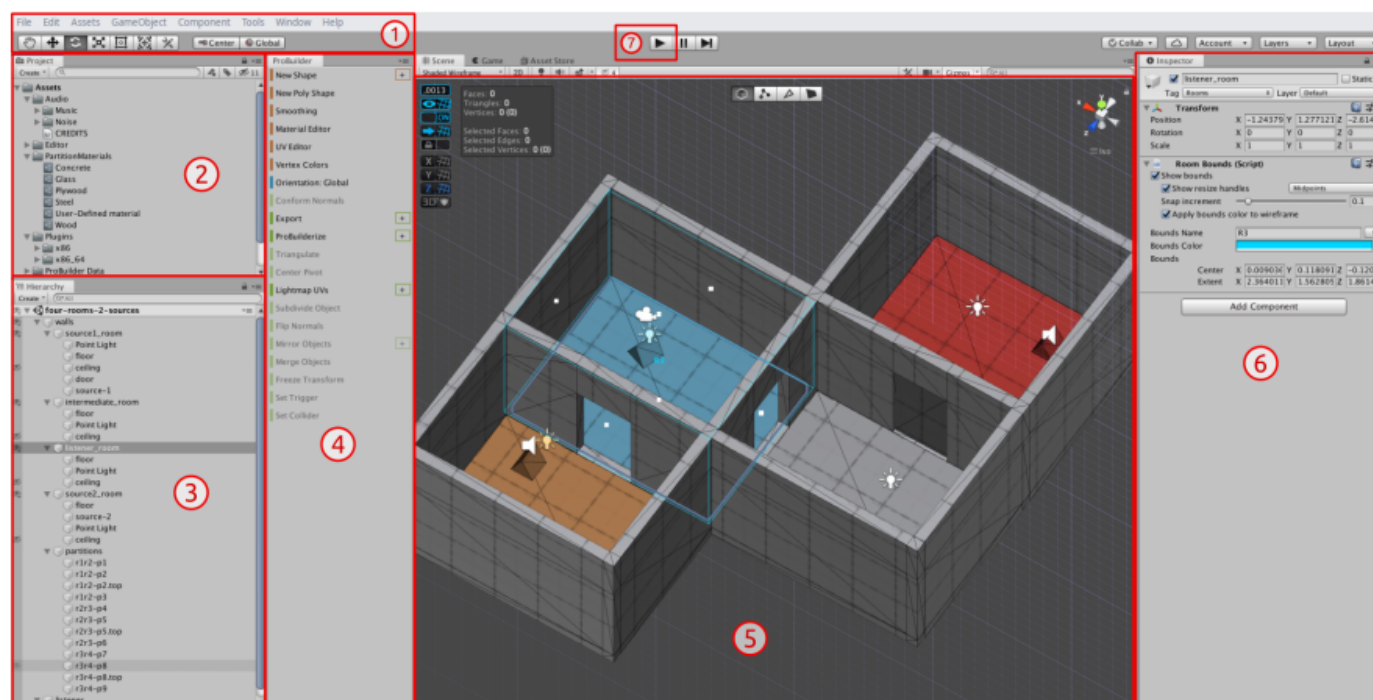
function get_all_paths_DFS(room_from, target_room):
    mark room_from as visited
    paths = []
    room_stack = { room_from.get_partitions() }
    while room_stack is not empty:
        partition_queue = room_stack.peek()
        if partition_queue is empty:
            mark current_room as not visited
            room_stack.pop()
            continue

        partition = partition_queue.get_next()
        adjacent_room = partition.get_adjacent_room()
        if adjacent_room is not set:
            continue
        if adjacent_room is visited:
            continue

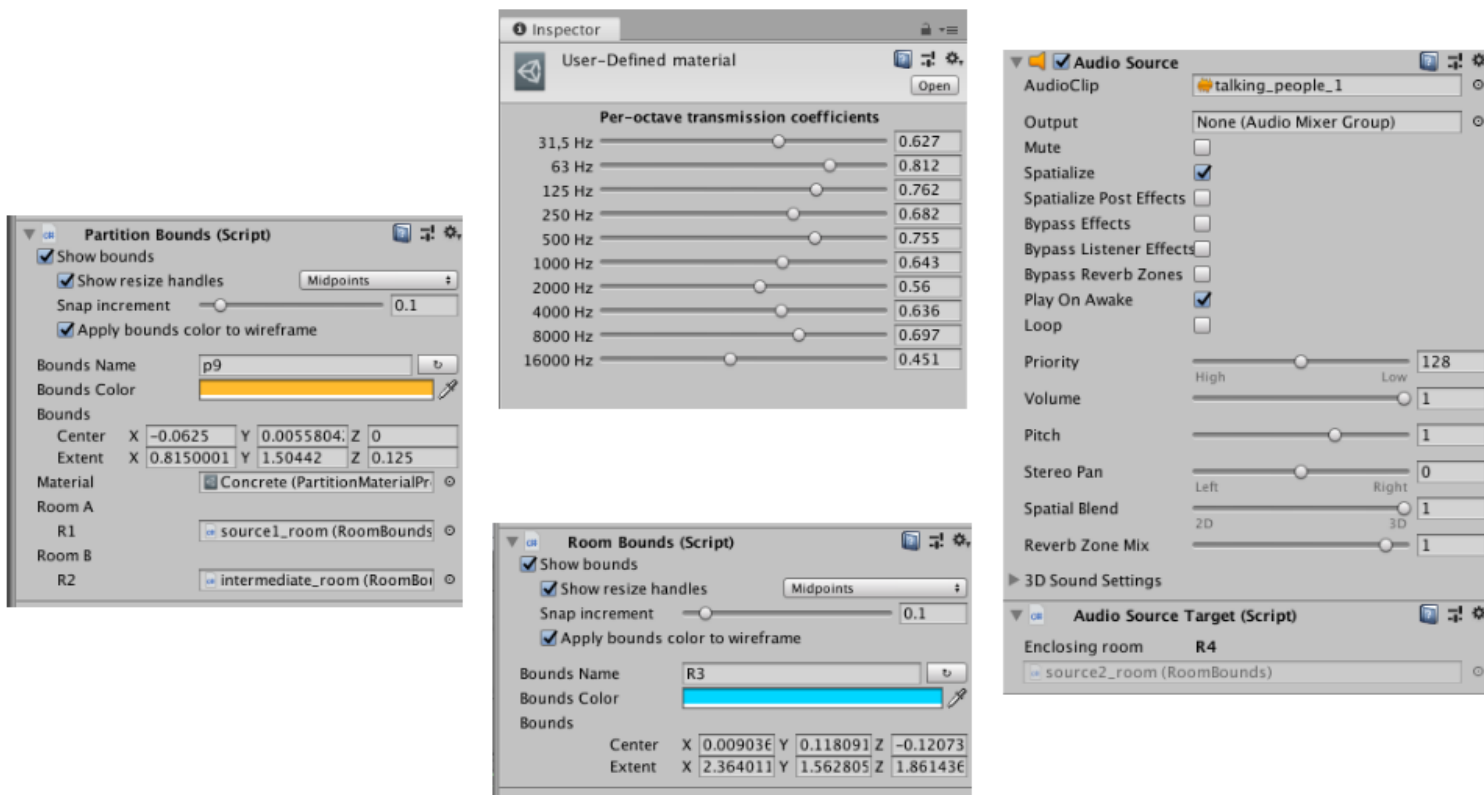
        if adjacent_room is target_room:
            build current_path from room_stack
            paths.add(current_path)
            continue

        mark adjacent_room as visited
        room_stack.push(room_from.get_partitions())
    return paths
  
```

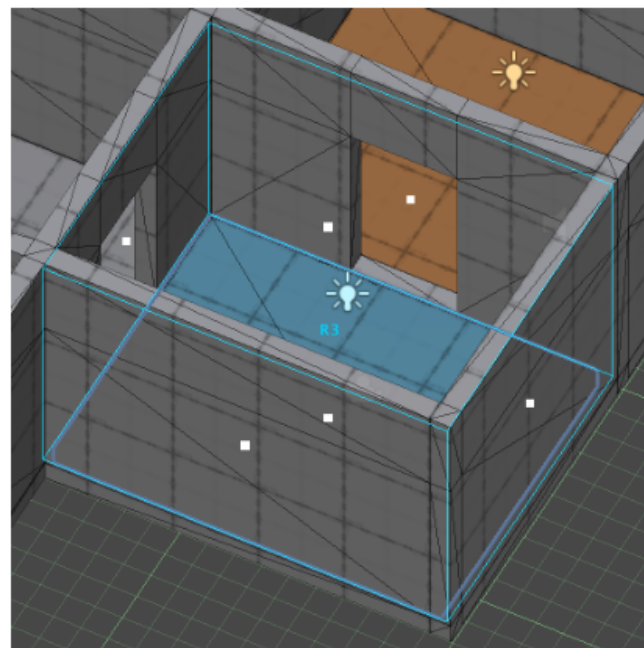
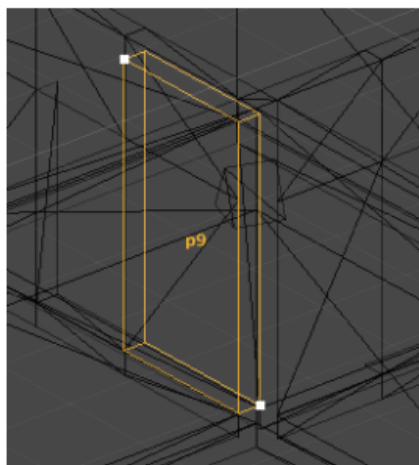
Інтерфейс редактора Unity



Розширення інтерфейсу інспектора об'єктів



Додаткові елементи у вікні сцени



Бізнес-модель стартап-проєкту

Проблема	Рішення	Унікальна ціннісна пропозиція	Прихована перевага	Споживачі
<ul style="list-style-type: none">■ відсутня або низька наочність автоматизованих рішень■ висока вартість людських послуг■ людський фактор	програмний продукт, здатний відтворювати поширення повітряного шуму в межах множини приміщень	оцінка ефективності шумоізоляції приміщення для найбільш інтуїтивним способом – на слух	органічна інтеграція в робочі процеси проєктування та реконструкції будівель	компанії, що виконують проєктування та реконструкцію будівель
	Ключові показники		Канали	
	кількість проданих ліцензій		відділи інтеграції компаній-споживачів	
Структура витрат			Потоки доходів	
<ul style="list-style-type: none">■ науково-дослідні та дослідно-конструкторські роботи■ оплата праці персоналу■ утримання робочих місць персоналу та виробничої інфраструктури■ обладнання, ліцензії програмного забезпечення для розробки■ податкові витрати			<ul style="list-style-type: none">■ доходи від продажу ліцензій програмного забезпечення та бібліотек матеріалів■ надання підтримки програмного забезпечення	

Фінансовий план стартап-проєкту



Одиниці виміру: тис. доларів США

Півріччя	Загальні витрати	Оплата праці	Підсумок витрат	Заплановані доходи	Прибуток (без оподаткування)
1	30	62	102	0	-102
2	50	112	62	245	83



Науково-інноваційна новизна

Розроблено спосіб ауралізації поширення повітряного шуму в будівлі, перевагою якого є простота реалізації та висока ефективність.

Розроблений спосіб може використовуватися для перетворення аудіосигналу в реальному часі з мінімально можливою втратою реалістичності звучання. Для зменшення кількості обчислень враховуються лише прямі шляхи поширення шуму без виникнення дифракції та реверберації, а перегородки-випромінювачі звуку вважаються вторинними точковими джерелами.



Висновки

1. Досліджено загальні принципи ауралізації та розроблено спосіб ауралізації поширення повітряного шуму в межах заданої множини приміщень, який дозволяє виконувати перетворення аудіосигналу в реальному часі з мінімально можливою втратою реалістичності.
2. Виконано розроблення програмної реалізації способу з використанням програмного рушія Unity та аналіз отриманих результатів.
3. Проведено оцінку комерційної життєздатності продукту: побудована бізнес-модель стартап-проєкту, розраховано прогнозовані показники потенційної прибутковості.



Апробування отриманих результатів

1. XII наукова конференція магістрантів та аспірантів
“Прикладна математика та комп’ютинг” ПМК-2019

Результати перевірки тексту дисертації на предмет плагіату



Submission author:
Сулема Євгенія Станіславівна

Check ID:
1000790203

Check date:
17.12.2019 21:49:04 GMT+0

Check type:
Doc vs Internet + Library

Report date:
17.12.2019 21:50:53 GMT+0

User ID:
83363

File name: 2019_Master_Glinskyi_2

File ID: 1000796230 Page count: 65 Word count: 12894 Character count: 97093 File size: 1.09 MB

12.1% Matches

Highest match: 6.54% with source <http://confopcb.iee.kpi.ua/proc/article/download/149178/148279>

9.2% Internet Matches 94 Page 67

5.27% Library matches 242 Page 69

0.69% Quotes

Quotes 5 Page 70

No references found

0% Exclusions

No exclusions found

Replacement

Character replacement 29



Дякую за увагу